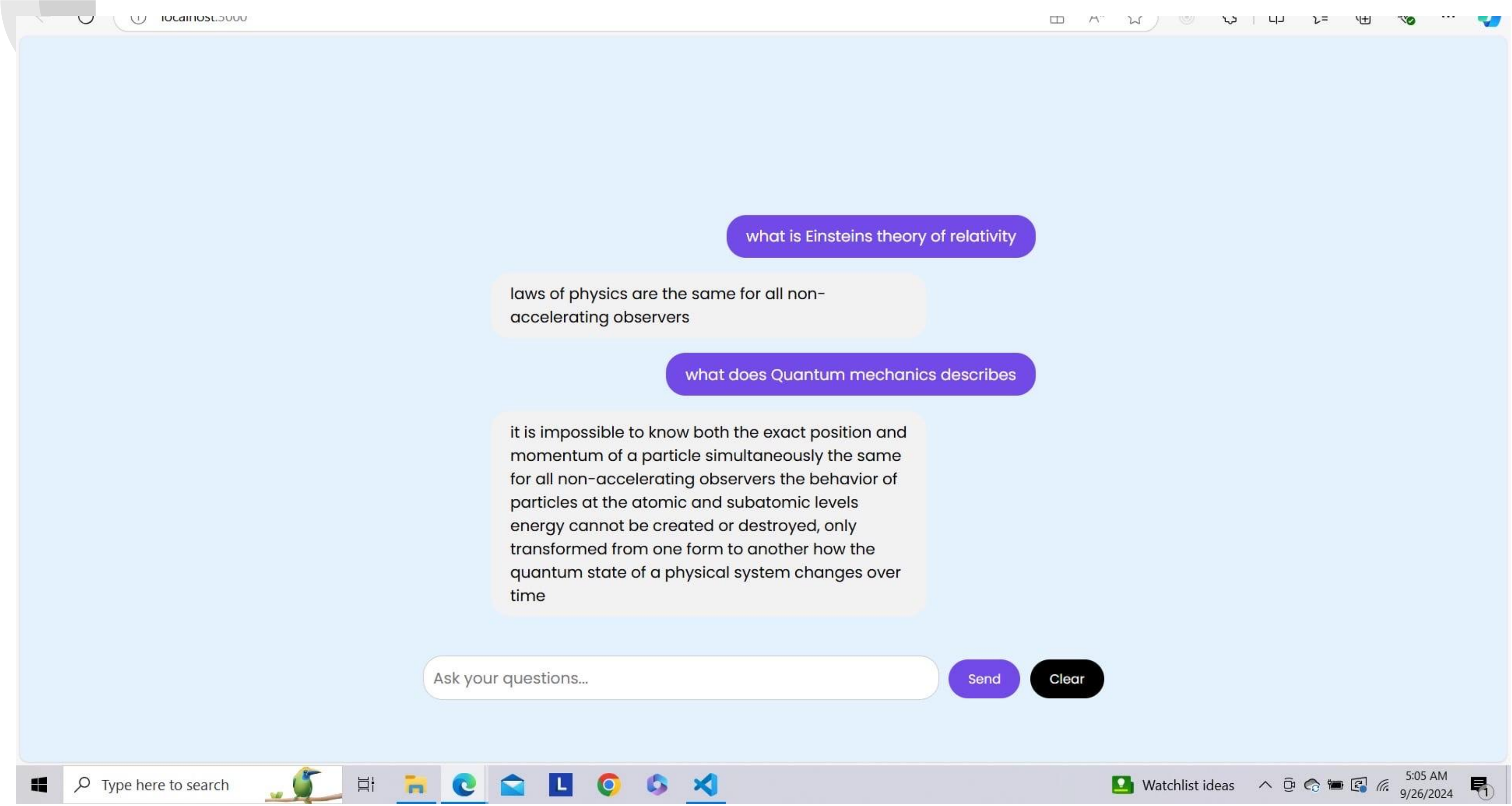
Milestone 1

By Sumaiya Jahan Sharlin(2011126042) &
Shaira Aktar Shove (2122179642)

Understanding the problem

Develop a physics-focused chatbot with the to content from the provided Physics Textbook.
The chatbot should be user-friendly, with a clean chat interface and support for clearing the chat.

Proposed User Interface



A comparative analysis of different Small/Large Language Models

Feature	Small Language Models (SLMs)	Large Language Models (LLMs)
Examples	DistilBERT, MiniLM	GPT-4, BERT-large, T5-large
Model Size	Small (e.g., 6M - 60M parameters)	Large (e.g., billions of parameters)
Performance	Suitable for specific or narrow tasks	Better at handling complex and open-ended tasks
Computation & Speed	Faster inference, uses less memory	Slower inference, requires more computing power
Training Costs	Lower training cost due to fewer parameters	Expensive to train, requiring significant hardware
Use Cases	Simple, real-time applications, mobile devices	Advanced NLP tasks like text generation, summarization
Fine-tuning Requirements	Requires less data and resources to fine-tune	Needs large datasets and more time for fine-tuning
Accuracy	Adequate for basic queries but less accurate on complex tasks	Higher accuracy, especially in understanding nuanced questions
Deployment	Easier to deploy, even on smaller devices	Typically deployed in cloud environments due to resource needs

For our limited domain (Provided Physics textbook), an SLM will offer a balance between speed and accuracy.

Small Language Models (SLMs) are better suited for this physics chatbot project because:

1. Focused Task: The chatbot is limited to a physics textbook. SLMs handle specific, domain-focused tasks efficiently, while LLMs are overkill for such narrow topics.
2. Speed and Efficiency: SLMs are faster and use fewer resources, providing quick responses, which is essential for real-time interaction.
3. Cost-Effective: SLMs are cheaper to deploy and maintain. LLMs, with their size and complexity, are more costly and not needed for this task.
4. Good Accuracy: SLMs like DistilBERT can handle the level of complexity in the textbook without sacrificing quality.
5. Easier Fine-Tuning: SLMs can be fine-tuned easily on the specific physics content, whereas LLMs require more effort and resources.

In short, SLMs are more efficient, affordable, and sufficient for this project's needs compared to LLMs.

What is an Embedding Model?

Embedding models are tools used in natural language processing (NLP) to convert words, phrases, or sentences into numerical vectors (arrays of numbers) so that computers can understand.

Embedding Model	What It Does	Strengths	Limitations	Suitability for Physics Chatbot
Word2Vec	Creates word-level embeddings from context	Fast and efficient	Lacks sentence context; struggles with multi-word meanings	Not suitable; misses crucial physics concepts
GloVe	Generates word embeddings based on global word statistics	Captures relationships between words	Focuses on word-level; no sentence understanding	Not suitable; fails to grasp contextual meaning
BERT	Provides deep contextual embeddings for tokens	Excellent at understanding context	Requires combining token embeddings for sentences	Not optimal for fast retrieval
FastText	Considers subword information for better coverage	Handles rare words well	Still focuses on word-level; limited context	Not suitable; doesn't capture full sentence meaning
InferSent	Produces sentence embeddings using supervised learning	Good for basic sentence-level tasks	Less powerful than newer models	Limited effectiveness for complex physics queries
USE (Universal Sentence Encoder)	Generates embeddings for sentences and paragraphs	Good for general NLP tasks	Not specialized for specific semantic searches	May not provide the precision needed for physics

SBERT (Sentence-BERT) is the best choice for our physics chatbot due to its ability to generate high-quality sentence embeddings, fast retrieval capabilities, and effective handling of context, making it ideal for answering physics-related questions accurately and efficiently.

What do we mean by Chunking Strategy?

Chunking strategy refers to the method of breaking down text into smaller, manageable pieces or "chunks." This is important in natural language processing (NLP) because it helps models understand and process the text more effectively.

A comparative analysis of different chunking strategies

Chunking Strategy	Description	Use Cases	Strengths	Limitations
Sentence Chunking	Divides text into individual sentences	Sentiment analysis, question answering	Simple and straightforward for sentence-level tasks	May miss context across sentences
Phrase Chunking	Dividing text into phrases based on grammatical structures	Information extraction, parsing	Captures relationships between words	Requires complex parsing techniques
Fixed-Size Chunking	Splits text into chunks of a specific size	Training models needing uniform input sizes	Ensures consistency in input size	May cut off important context
Overlapping Chunking	Creates chunks that overlap with each other	Contextual analysis, dialogue systems	Retains context across chunk boundaries	Increased complexity in handling overlaps
Semantic Chunking	Breaking text into chunks based on the meaning of the words and phrases	Ensures coherent context per chunk	Captures complex relationships.	Requires deep semantic understanding.
Hierarchical Chunking	Organizes chunks in a hierarchical structure	Document summarization, topic modeling	Maintains context at multiple levels	More complex to implement and analyze

Semantic Chunking is the best choice for your physics chatbot as it enhances the understanding of complex physics concepts, preserves context, and improves the overall accuracy of responses.