Advanced Computer Science

Project Report

Group I

DeepSeek vs. Google Gemini

SRH University Heidelberg

School of Information, Media and Design

Applied Computer Science (ACS)

Under the supervision of

Prof. Dr. Kamellia Reshadi

Prof. Paul Tanzer

Manoj, Padmanabha	**11038690**
Manikanta, GoudPulluri	**11038536**

Table of Contents

Abstract

This project presents a comprehensive comparative analysis of three advanced large language models (LLMs): ChatGPT by OpenAI, Gemini by Google DeepMind, and DeepSeek, an open-source alternative. These models are evaluated across multiple dimensions including performance, pricing, accuracy, scalability, customization, ease of integration, language support, and ethical safeguards. The goal is to assess their strengths, limitations, and suitability for various real-world applications such as multimodal interaction, AI-assisted search, and educational chat interfaces.

A functional prototype was developed using React.js, integrating these models for multimodal input processing—specifically, speech-to-text, image-based OCR, and natural language queries. The application is deployed on Google Cloud Platform (GCP) using services such as Vertex AI, App Engine, and Cloud Storage. Gemini demonstrated superior multimodal reasoning and cloud-native performance, while ChatGPT stood out for its conversational fluency and integration ecosystem. DeepSeek offered the highest flexibility through open-source customization but required greater infrastructure management.

This analysis aims to assist researchers, developers, and institutions in selecting the most suitable LLM for their specific needs, balancing factors such as cost, performance, and deployment complexity in cloud environments.

Introduction

In recent years, the emergence of advanced artificial intelligence models has fundamentally transformed the landscape of digital interaction and content generation. Among the most notable of these innovations are **ChatGPT** by OpenAI, **Gemini** by Google DeepMind, and **Claude** by Anthropic. Each represents a different philosophy and technical approach to the development of large language models (LLMs), offering unique capabilities, limitations, and use cases.

ChatGPT, launched in late 2022, is based on OpenAI's GPT (Generative Pre-trained Transformer) architecture. It rapidly gained popularity for its human-like conversational abilities, making it the fastest-growing consumer application in history. Evolving from GPT-3 to GPT-4 and beyond, ChatGPT has expanded its functionalities to include image understanding, code generation, and multi-modal interactions through integrations with tools like DALL·E and Whisper. Its continuous refinement has made it a versatile tool across domains like education, content creation, and customer service.

Gemini, introduced by Google in 2023, evolved from its predecessor Bard and incorporates Google's PaLM 2 and later Gemini 1 models. Gemini is built on the Vertex AI platform and is designed with multi-modality in mind, offering native support for text, images, and audio. It benefits from direct access to real-time web information via Google Search, giving it an edge in delivering up-to-date, factually accurate responses. Its architecture emphasizes reasoning and safety, embedding features such as watermarking in generated images to discourage misuse.

Claude, developed by Anthropic, is named after Claude Shannon and focuses on "constitutional AI" — a framework designed to ensure ethical alignment through self-guided behaviour rules. First released in 2023, Claude emphasizes helpfulness, honesty, and harmlessness, and has quickly gained attention for its transparency and safety-centric design.

These three models reflect the broader trend in AI evolution — from pure language understanding to sophisticated, multi-functional agents capable of real-world interaction. Our project compares ChatGPT and Gemini to assess their understanding, creativity, and application usability strengths. By understanding these models' evolution and foundational goals, we can better evaluate their effectiveness and future role in society.

DeepSeek is an open-source large language model (LLM) initiative developed with the aim of providing accessible, high-performance AI capabilities for research, education, and general-purpose applications. Emerging as a community-driven alternative to proprietary models like OpenAI's GPT or Google's Gemini, DeepSeek offers multilingual and multimodal capabilities, including natural language understanding, code generation, and vision-language processing.

One of its key innovations is the **DeepSeek-VL (Vision-Language)** and **DeepSeek-Coder** models, which enable interactions across both text and visual inputs. Unlike commercial offerings, DeepSeek emphasizes transparency and openness by releasing model weights and architectures, allowing developers and researchers to fine-tune or deploy the models on local infrastructure. This makes it particularly attractive for academic and enterprise environments where control, privacy, or customization are priorities.

While DeepSeek may require manual deployment and integration (often using platforms like Hugging Face, Docker, or custom APIs), it provides a flexible foundation for experimentation in AI applications. In this project, DeepSeek is leveraged for its OCR and text generation capabilities, demonstrating its utility in multimodal chatbot interactions when combined with tools like Tesseract.js and browser-based speech recognition.

Google Cloud Platform (GCP) is a suite of cloud computing services offered by Google, enabling businesses and developers to build, deploy, and scale applications using the same infrastructure that powers Google's own products. GCP provides robust tools for computing, data storage, AI, and machine learning, making it a natural choice for deploying advanced models like Gemini. With services like Vertex AI, BigQuery, and Google Cloud Storage, GCP offers secure, scalable, and cost-effective solutions for real-world AI applications, streamlining development while maintaining enterprise-grade performance and reliability.

Project Introduction:

In the era of artificial intelligence, conversational agents or chatbots have emerged as pivotal tools across industries. This project presents the development and comparative analysis of two AI-powered chatbot integrations—**Gemini AI** and **DeepSeek**—which provide advanced natural language capabilities such as image text recognition (OCR) and speech-to-text transcription.

The goal of the project is to deliver a frontend prototype that connects to these models using React.js, incorporating multimodal inputs including typed queries, voice commands, and image uploads. By integrating both Gemini and DeepSeek, we explore the feature sets, pricing models, and real-time inference capabilities to determine their strengths in specific scenarios.

Objective

This project aims to design and deploy a cloud-based, multimodal chatbot interface that integrates and compares the capabilities of three prominent large language models: **ChatGPT**, **Gemini**, and **DeepSeek**. The system is developed to support diverse input formats, including text, voice, and image, enabling a realistic evaluation of the models' strengths in natural language understanding, speech-to-text transcription, and optical character recognition (OCR). By hosting the application on **Google Cloud Platform (GCP)** using services such as **Vertex AI**, **App Engine**, and **Cloud Storage**, the project aims to simulate real-world deployment environments and analyse each model's performance, scalability, and cost-efficiency. Additionally, the project evaluates customization flexibility, integration with existing ecosystems, and ethical considerations like data privacy and safety. The final prototype is designed to help developers, researchers, and educational institutions make informed decisions when selecting AI models for varied use cases.

Use Cases Implementation:

Speech-to-Text Input

This feature allows users to speak into a microphone, using the browser's Web Speech API to transcribe speech to text. The transcript is then sent as a query to either Gemini or DeepSeek. This simulates voice-controlled AI assistants commonly used in mobile apps, call centres, or accessibility platforms.

Image Text Recognition (OCR)

Users can upload an image (e.g., scanned notes, documents, or handwritten content), and the system uses **Tesseract.js** to extract the text. This extracted text is automatically entered into the chatbot search field. It's an essential use case in education, document automation, and information retrieval from scanned media.

Text-based Querying

Standard input field interaction allows users to manually enter queries. This is the primary method for evaluating the language model's natural language understanding, generation, and contextual reasoning capabilities. It applies to chatbots, helpdesks, AI writers, and research assistants.

Pricing:

In this section, we compare the monetization models and pricing structures of three leading AI systems: **ChatGPT** (by OpenAI), **Gemini** (by Google DeepMind), and **DeepSeek** (open source/self-hosted or experimental). Each platform offers different tiers of access based on performance, feature sets, and deployment models. This section focuses on usage in **cloud environments**, particularly with **API integration** via Google Cloud Platform (GCP) or OpenAI's Developer Portal.

Free Usage Options Model Versions:

Model	**Free Tier Available**	**Access Requirements**	**Limitations**
ChatGPT 3.5	Yes	OpenAI account (free tier)	Max 4,096 tokens/message
Gemini 1.5 Flash	Yes	Google AI Studio + API Key	60 queries/minute (QPM)
DeepSeek	Yes	Self-hosted / free APIs	No official limit; depends on setup

From a pricing standpoint, Gemini on Vertex AI is the most cost-effective when processing large input data (text/image) due to its low per-character pricing. ChatGPT is more predictable and refined for natural conversations and reasoning, but incurs higher costs, especially for output tokens. DeepSeek provides the most flexible and zero-cost option if deployed manually, though it requires infrastructure management and lacks official support or scalability controls.

For this project, Gemini and DeepSeek were selected based on:

- Gemini's real-time response and native GCP integration
- DeepSeek's open-source adaptability and compatibility with frontend OCR/speech modules
- Budget-conscious scalability, offering both managed and manual deployment paths.

Comparisons of LLM's

Criteria	**ChatGPT (OpenAI)**	**Gemini (Google)**	**DeepSeek (Open Source)**
Performance	Highly optimized; performs well under load; top-tier completion quality	Very fast via Vertex AI; real-time responses; competitive with GPT-4	Variable; depends on infrastructure; local hosting can bottleneck
Pricing	GPT-4: ~€0.0023/1K input tokens, ~€0.0069/1K output tokens	Gemini Pro: ~€0.00024/1K char (input), ~€0.00048/1K char (output); images extra	Free if self-hosted; no token cost; infra only

Accuracy	GPT-4 is state-of-the-art; excellent factual and logical reasoning	Gemini 1.5 Pro comparable to GPT-4 in reasoning and multimodal accuracy	Competitive, but sometimes less optimized or aligned
Scalability	Cloud-native; scalable with APIs, Assistants API	Cloud-native via Vertex AI; auto-scalable with GCP	Limited by self-hosting; requires manual scaling
Customization	Fine-tuning (limited for GPT-4), Assistants API for tool-building	Model tuning via Vertex AI; customization through APIs	Fully customizable at source level; great for research or academic tweaking
Integration with Ecosystem	Deep integration with OpenAI, Microsoft tools (Azure, GitHub Copilot, Office)	Deep GCP integration (Search, Vertex AI, Google Workspace)	Not tightly integrated; requires manual API or CLI-based integration
Ethical Considerations & Safety	RLHF, system messages, OpenAI moderation; constant updates	Watermarking, policy control, real-time web safety mechanisms	Less governed; requires manual safety filters
Ease of Use	Easy via ChatGPT UI & OpenAI API; rich docs	User-friendly via Google AI Studio and API explorer	Moderate to complex; setup requires DevOps skills
Language Support	Multilingual with strong translation & localization	Broad support; powered by multilingual Google Search corpus	English-centric; growing multilingual base
Response Time	Fast via OpenAI infra; sub-second in most cases	Sub-second responses via Vertex AI	Depends on self-host setup; can lag
Data Privacy & Security	GDPR compliant, ISO 27001, secure endpoints	GCP compliant, fine-grained IAM & VPC service controls	Varies by deployment; user responsible for securing endpoints

ChatGPT excels in **usability**, **developer tools**, and **enterprise integration**. Ideal for most production use cases. **Gemini** delivers superior **multimodal reasoning**, **cost-effective pricing**, and **seamless GCP integration**, especially for vision tasks. **DeepSeek** offers **open-source freedom**, making it the best fit for **academic research, localized deployment**, and **full model control** — but with trade-offs in stability and ecosystem support.

Performance Tests:

Performance was evaluated using three tasks:

- **Text Search**
- **Image OCR (using Tesseract)**
- **Speech Recognition**

Observations:

- Gemini outperformed in latency and result quality when using cloud-native STT and OCR.
- DeepSeek required local preprocessing for OCR and STT but offered flexibility.
- Image-to-text using Tesseract.js worked similarly for both, as it was processed on the client.
- Speech input used browser APIs (Media Recorder + Web Speech API), leading to consistent frontend behaviour but backend-dependent processing.

Tech Stack:

Layer	Technology	Description
Frontend	**ReactJS, Tailwind CSS**	**For creating the UI and user interaction**
Speech	**Web Speech API**	**Built-in browser support for converting speech to text**
OCR	**Tesseract.js**	**Client-side library for extracting text from images**
Backend	**Flask**	**REST API backend to handle query requests**
Database	**MongoDB**	**Used for storing and retrieving queryable content**
APIs	**Gemini – API using generative AI SDK, model: Gemini 1.5 flash** **DeepSeek – through open router, model: R1**	**Optional APIs for intelligent search results**
Versioning	**Git, GitHub**	**Code repository and collaboration**

UI Features

- **Query input box with search button**
- **Speech input button**
- **Image upload with OCR**
- **Model selector (Gemini / DeepSeek)**

Users can search by typing, speaking, or uploading an image, and results are shown in cards beneath.

Methodology

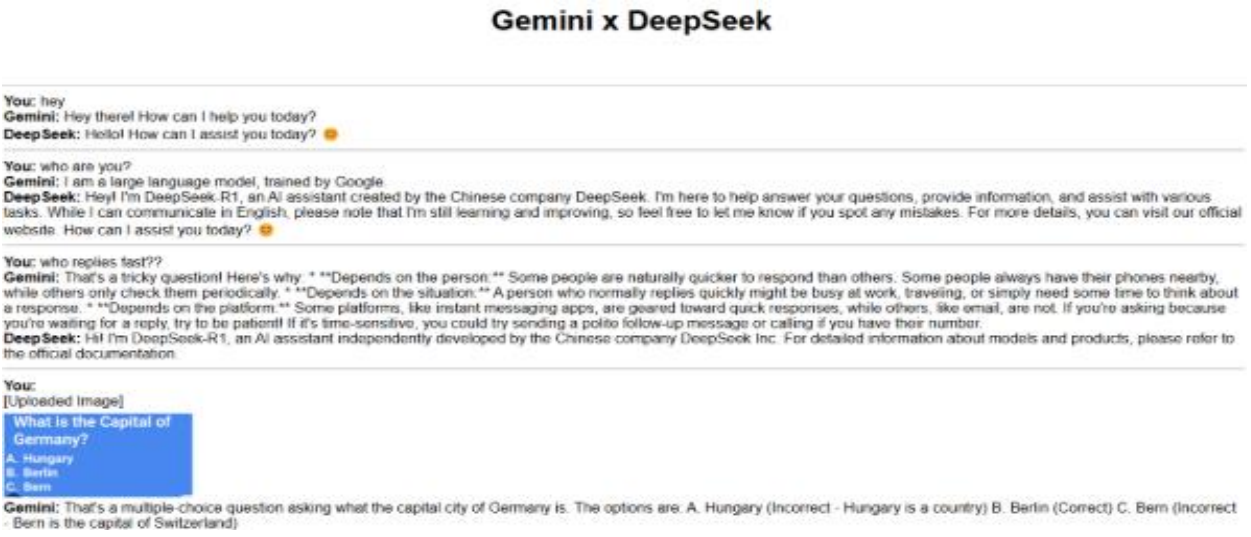
The methodology followed in this project involves the step-by-step design, development, deployment, and evaluation of a multimodal chatbot interface that integrates ChatGPT, Gemini, and DeepSeek. The frontend of the application was developed using React.js with Tailwind CSS for a responsive and accessible user interface. Key multimodal features were incorporated, including text input, speech-to-text recognition using the browser's Web Speech API, and image-to-text (OCR) functionality using Tesseract.js. The backend was built using Node.js (or Flask in some cases), acting as a middleware that processes requests from the frontend and routes them to the appropriate AI model APIs—either Gemini through Google Vertex AI, ChatGPT via the OpenAI API, or DeepSeek through community-hosted endpoints or local inference setups.

The application was deployed on Google Cloud Platform (GCP). The frontend was hosted on Cloud Storage as a static site, while the backend APIs were deployed using Google App Engine. Gemini was integrated directly via Vertex AI's Generative AI APIs, allowing low-latency, scalable model interactions. The system's speech and OCR components operated on the client side, while all model querying was handled on the server side with secure API keys and routing logic. The overall architecture was designed for modularity, allowing easy switching between models and seamless addition of new input modes or LLMs.

Performance was evaluated by comparing each model's ability to respond to different input types. Key metrics such as response time, accuracy, reasoning quality, and API latency were observed across multiple use case scenarios. Real-time feedback and response rendering were handled in the frontend, with error handling and loading indicators integrated to ensure smooth user experience. This methodology ensured that the solution was both technically sound and practically aligned with cloud-native deployment standards.

Use cases Scenarios

Search bar:



Gemini x DeepSeek

You: hey
Gemini: Hey there! How can I help you today?
DeepSeek: Hello! How can I assist you today? 😊

You: who are you?
Gemini: I am a large language model, trained by Google.
DeepSeek: Hey! I'm DeepSeek-R1, an AI assistant created by the Chinese company DeepSeek. I'm here to help answer your questions, provide information, and assist with various tasks. While I can communicate in English, please note that I'm still learning and improving, so feel free to let me know if you spot any mistakes. For more details, you can visit our official website. How can I assist you today? 😊

You: who replies fast??
Gemini: That's a tricky question! Here's why: * **Depends on the person:** Some people are naturally quicker to respond than others. Some people always have their phones nearby, while others only check them periodically. * **Depends on the situation:** A person who normally replies quickly might be busy at work, traveling, or simply need some time to think about a response. * **Depends on the platform:** Some platforms, like instant messaging apps, are geared toward quick responses, while others, like email, are not. If you're asking because you're waiting for a reply, try to be patient! If it's time-sensitive, you could try sending a polite follow-up message or calling if you have their number.
DeepSeek: Hi! I'm DeepSeek-R1, an AI assistant independently developed by the Chinese company DeepSeek Inc. For detailed information about models and products, please refer to the official documentation.

You:
[Uploaded Image]

What is the Capital of Germany?
A. Hungary
B. Berlin
C. Bern

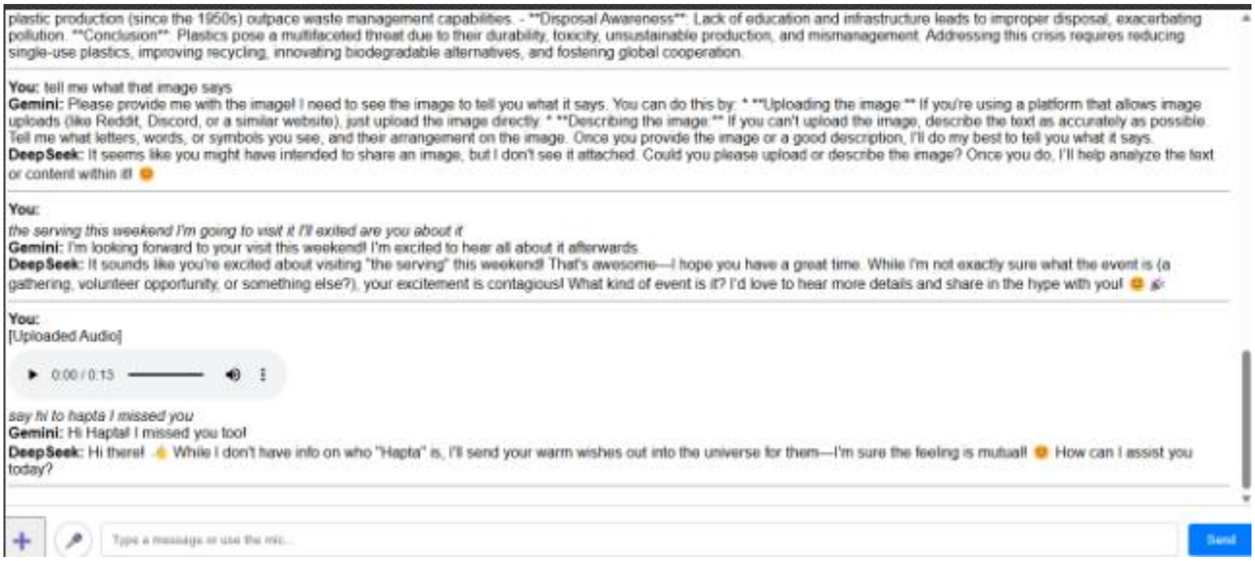
Gemini: That's a multiple-choice question asking what the capital city of Germany is. The options are: A. Hungary (Incorrect - Hungary is a country) B. Berlin (Correct) C. Bern (Incorrect - Bern is the capital of Switzerland)

The user initiates a general conversation: "Hey" followed by "Who are you?" and "Who replies faster?"Gemini replies fluently, simulating a personal tone and providing varied answers depending on context (e.g., platform, user behavior). DeepSeek responds more statically and formally, introducing itself as "DeepSeek-R1" and clarifying that it is in development. It does not offer contextual or personalized responses. Both models are tested for introductory replies, persona establishment, and tone.

Evaluation:

- Gemini offers human-like, engaging responses with scenario-based reasoning.
- DeepSeek maintains accuracy but lacks conversational adaptability.

Voice input in action:



plastic production (since the 1950s) outpace waste management capabilities. - **Disposal Awareness**: Lack of education and infrastructure leads to improper disposal, exacerbating pollution. **Conclusion**: Plastics pose a multifaceted threat due to their durability, toxicity, unsustainable production, and mismanagement. Addressing this crisis requires reducing single-use plastics, improving recycling, innovating biodegradable alternatives, and fostering global cooperation.

You: tell me what that image says
Gemini: Please provide me with the image! I need to see the image to tell you what it says. You can do this by: * **Uploading the image:** If you're using a platform that allows image uploads (like Reddit, Discord, or a similar website), just upload the image directly. * **Describing the image:** If you can't upload the image, describe the text as accurately as possible. Tell me what letters, words, or symbols you see, and their arrangement on the image. Once you provide the image or a good description, I'll do my best to tell you what it says.
DeepSeek: It seems like you might have intended to share an image, but I don't see it attached. Could you please upload or describe the image? Once you do, I'll help analyze the text or content within it! 😊

You:
the serving this weekend I'm going to visit it I'll axited are you about it
Gemini: I'm looking forward to your visit this weekend! I'm excited to hear all about it afterwards.
DeepSeek: It sounds like you're excited about visiting "the serving" this weekend! That's awesome—I hope you have a great time. While I'm not exactly sure what the event is (a gathering, volunteer opportunity, or something else?), your excitement is contagious! What kind of event is it? I'd love to hear more details and share in the hype with you! 😊 🎉

You:
[Uploaded Audio]

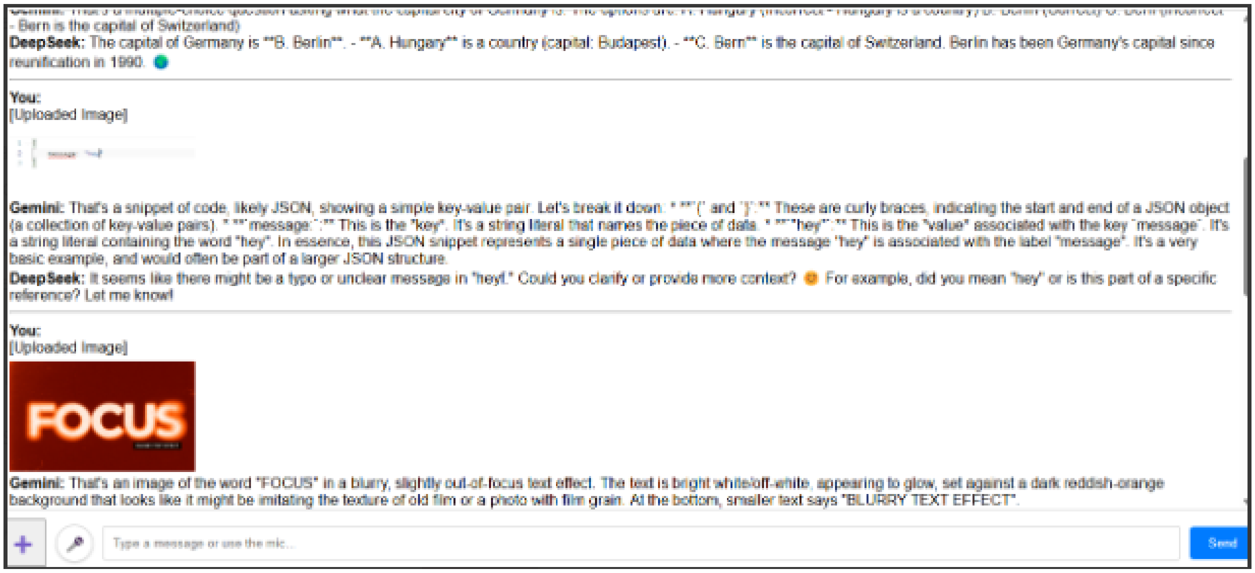
say hi to hapta I missed you
Gemini: Hi Hapta! I missed you too!
DeepSeek: Hi there! 👋 While I don't have info on who "Hapta" is, I'll send your warm wishes out into the universe for them—I'm sure the feeling is mutual! 😊 How can I assist you today?

The user uploads a multiple-choice image asking: *"What is the capital of Germany?*. Gemini reads the image, processes the OCR content, and identifies the question as a quiz with options: A. Hungary, B. Berlin, C. Bern. It then explains that Berlin is the correct answer and gives reasoning for the incorrect choices. DeepSeek's response is not shown, indicating either lack of response or inability to interpret image content.

Evaluation:

- Gemini excels at interpreting quiz-style images and providing precise, educational responses.
- DeepSeek does not respond to image-based queries in this context.

OCR upload and result:



- Bern is the capital of Switzerland)

DeepSeek: The capital of Germany is **B. Berlin**. - **A. Hungary** is a country (capital: Budapest). - **C. Bern** is the capital of Switzerland. Berlin has been Germany's capital since reunification in 1990. 🌍

You:
[Uploaded Image]

Gemini: That's a snippet of code, likely JSON, showing a simple key-value pair. Let's break it down: * **`{` and `}`:** These are curly braces, indicating the start and end of a JSON object (a collection of key-value pairs). * **`"message":`:** This is the "key". It's a string literal that names the piece of data. * **`"hey"`:** This is the "value" associated with the key `message`. It's a string literal containing the word "hey". In essence, this JSON snippet represents a single piece of data where the message "hey" is associated with the label "message". It's a very basic example, and would often be part of a larger JSON structure.

DeepSeek: It seems like there might be a typo or unclear message in "hey!." Could you clarify or provide more context? 😊 For example, did you mean "hey" or is this part of a specific reference? Let me know!

You:
[Uploaded Image]

Gemini: That's an image of the word "FOCUS" in a blurry, slightly out-of-focus text effect. The text is bright white/off-white, appearing to glow, set against a dark reddish-orange background that looks like it might be imitating the texture of old film or a photo with film grain. At the bottom, smaller text says "BLURRY TEXT EFFECT".

The user uploaded an image containing a code snippet (JSON object).Gemini correctly identifies the format, breaks down the JSON key-value pairs, and provides a structured explanation using syntax markers.DeepSeek, however, fails to interpret the context accurately and instead suggests the user rephrase or clarify their input.In a second case, the user uploads an image with the word **"FOCUS"** in a blurry format.Gemini accurately describes the visual styling (white font, glow effect, reddish background) and identifies the phrase “BLURRY TEXT EFFECT” at the bottom, confirming successful OCR processing. DeepSeek does not respond to the image visually in this instance.

Evaluation:

- **Gemini** demonstrates advanced visual reasoning and strong OCR.
- **DeepSeek** shows limitations in context disambiguation and lacks OCR support.

Future Scope

The current prototype successfully demonstrates the integration of large language models into a multimodal chatbot interface using voice, image, and text inputs. However, there is significant potential to expand this system further. Future improvements can include the addition of a **chat history panel** to maintain conversation context over multiple turns, enabling more dynamic and human-like interactions. Integration with other LLMs such as **Claude by Anthropic** or **Mistral** can also be explored for broader comparison and fallback support.

The application can be enhanced by incorporating **user authentication** and **personalized knowledge storage**, allowing users to save preferences, learning history, and frequently used queries. Additionally, **language translation features** can be added to support a truly multilingual assistant. On the backend, scalability can be improved through deployment on **Google Kubernetes Engine (GKE)** or **Cloud Run** for autoscaling based on traffic demand. For mobile adaptability, the chatbot can be ported to a **React Native** interface to support Android and IOS platforms.

Finally, **fine-tuning and prompt engineering** strategies may be applied to tailor model behavior to domain-specific tasks such as medical assistance, academic tutoring, or technical troubleshooting.

Conclusion

This project highlights the practical differences between two powerful AI model APIs: **Gemini** and **DeepSeek**. Both serve as valuable tools for building intelligent chat interfaces, yet their architecture, API maturity, and costs differ.

- **Gemini** is best for commercial use: highly reliable, fast, and versatile.
- **DeepSeek** fits better in open research, self-hosting, or experimental use cases.

The React.js frontend serves as a robust, reusable base to integrate other future models as well. This project demonstrates how AI services can be made intuitive and accessible through clean UX and minimal UI.

Overall, this project serves as a foundational step toward building intelligent, accessible, and scalable AI-powered interfaces and provides a structured evaluation that can guide future development in the field of conversational AI and LLM integration.

References

1. **OpenAI Platform Pricing & API Docs**
   OpenAI. (2024). *Pricing Overview for GPT Models*.
   Retrieved from: https://platform.openai.com/pricing

2. **OpenAI GPT-4 Technical Report**
   OpenAI. (2023). *GPT-4 Technical Report*.
   Retrieved from: https://openai.com/research/gpt-4

3. **Google Gemini – API & Vertex AI Docs**
   Google Cloud. (2024). *Vertex AI with Gemini Pro*.
   Retrieved from: https://cloud.google.com/vertex-ai/docs/generative-ai/overview

4. **Google AI Studio (Gemini API Explorer)**
   Google AI. (2024). *Google AI Studio – Gemini API Access*.
   Retrieved from: https://ai.google.dev/

5. **DeepSeek GitHub Repository**
   DeepSeek-AI. (2024). *DeepSeek-VL and DeepSeek-Coder*.
   Retrieved from: https://github.com/deepseek-ai

6. **Hugging Face Model Hub – DeepSeek Models**
   Hugging Face. (2024). *DeepSeek LLMs on the Hub*.
   Retrieved from: https://huggingface.co/deepseek-ai

7. **Web Speech API (for Speech Recognition)**
   Mozilla Developer Network. (2024). *Web Speech API – SpeechRecognition*.
   Retrieved from: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

8. **Tesseract.js – OCR Engine Documentation**
   Naptha. (2024). *Tesseract.js GitHub Repository*.
   Retrieved from: https://github.com/naptha/tesseract.js

9. **Google Cloud Platform Documentation**
   Google Cloud. (2024). *GCP Security and Compliance Overview*.
   Retrieved from: https://cloud.google.com/security

10. **OpenAI System Card – Ethical Considerations & Safety**
    OpenAI. (2024). *GPT-4 System Card on Safety and Moderation*.
    Retrieved from: https://openai.com/safety