Certified Cloud Applied Generative AI Engineer (GenEng)

Version: 6.0 (Implementation and adoption starting from Feb 1, 2024)

Executive Summary

The Cloud Applied Generative AI Engineering (GenEng) certification program aims to prepare individuals for the revolutionary era of Generative AI, which promises to transform industries and generate significant economic benefits. The one-year program combines onsite and online instruction, covering topics such as GenAI application development, cloud computing, and DevOps. Participants learn practical skills in TypeScript, Python, front-end development using Next.js, and GenAI-related technologies. The program emphasises real-world application, enabling participants to start earning through freelancing or other opportunities after the second quarter. At the end of the program, students can choose from various specialisations, including Web3, healthcare, finance, engineering, and more.

- The first quarter covers TypeScript and Python programming.
- The second quarter focuses on front-end development using Next.js.
- The third quarter introduces API development, database, containers, cloud deployment, and DevOps.
- The fourth quarter covers custom GPT development.
- The program allows students to specialise in one of several areas, such as Web3, Blockchain, and GenAI Integration or Healthcare and Medical GenAI.
- The program is a hybrid program with both online and on-site classes.
- The program has a focus on practical application and development in Generative AI, rather than pure model development.
- The program also offers students the opportunity to earn money during the program through freelancing or other work.
- The program is designed to prepare students for the rapidly changing landscape of AI and to help them start earning in the field as early as the second quarter.

The Details

Generative AI is poised to completely transform our lives and work landscape. McKinsey & Company estimates that generative AI could add $2.6 trillion to $4.4 trillion in economic benefits annually across various industries. This will be achieved through increased automation, improved decision-making, and personalised experiences. It is transformative for tech and jobs. It's critical, must-know knowledge across industries and businesses and if you don't keep up, you become obsolete in tech cycles that are moving fast. As new Gen AI-powered technologies keep coming and demand for skills are changing rapidly, workforce and professional training is exploding and is having difficulty in keeping up.

Our one-year GenEng certification program teaches you to get ready for this new revolutionary era. It consists of four quarters of three months each. At the completion of the program the students may choose to specialise in a specific area. It is a hybrid program consisting of onsite and online classes. It teaches you to develop smart applications using just GenAI, but also cutting-edge Web, and Cloud technologies. OpenAI Chat GPT 4, Google Gemini APIs, and Langchain for GenAI. TypeScript, React, Next.js, and Tailwind for the front end. Python, Numpy, Pandas, FastAPI, Pedantic, SQLAlchemy, and Postgresql databases for backend and API development. Containers (Docker), Google Cloud Run, Azure Container Apps, and Kubernetes for development, testing, and cloud deployment. The focus of the program is not on LLM model development but on applied Cloud GenAI Engineering (GenEng), application development, and fine-tuning of foundational models.

This program is an earn-as-you-learn program. The diligent and hardworking participants will be able to start earning after the second quarter in freelancing, template development, traditional jobs and remote work, and other front-end markets. After the second quarter, the participants will be enhancing their skills with GenAI, API, Cloud, and DevOps technologies.

Quarter 1: TypeScript and Python Programming
In the first quarter, we will learn the two most used programming languages in GenAI Application Development, TypeScript for User interfaces, and Python for Application Programming Interfaces (APIs). We will cover both functional and object-oriented paradigms. TypeScript programming will be taught onsite and Python programming online.

Quarter 2: Front-end Development using Next.js and TypeScript
In this quarter we will learn to build and deploy state-of-the-art web user interfaces using TypeScript 5, React 18, Next.js 14, and Tailwind CSS 3. We will use beautifully designed and customizable Shadcn UI React components which are built with Headless Radix UI and Tailwind CSS to speed up our development life cycle. Using

Headless CMSs will also be covered. We will also learn to use Vercel AI SDK, an open-source library for building AI-powered user interfaces. The quarter will end with you learning to deploy these UI apps on Vercel Cloud and CDN.

Quarter 3: API Design, Development, and Deployment using FastAPI, Containers, and OpenAPI Specifications
An API-as-a-Product is a type of Software-as-a-Service that monetizes niche functionality, typically served over HTTP. OpenAI APIs are themselves this kind of service. An application programming interface economy, or API economy, refers to the business structure where APIs are the distribution channel for products and services. In this quarter we will learn to develop APIs not just as a backend for our frontend but also as a product itself. In this model, the API is at the core of the business's value.

We will be using Python-based FastAPI as our core library and Pedantic, SQLAlchemy, and Postgresql databases for API development. Docker Containers will be our fundamental building block for development, testing, and deployment. For local development, we will be using Docker Compose and DevPod which is Dev-Environments-As-Code, for testing Pytest and Testcontainers, and for deployment Google Cloud Run, Azure Container Service, and Kubernetes. We will be using Terraform as our Infrastructure as Code (IaC) tool. OpenAI Chat GPT 4, Google Gemini APIs, and Langchain will be used to build these API-as-a-Product.

Quarter 4: Custom GPT and GPT Actions
In this Quarter we will learn to create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills. We will also learn how to build a GPT action that intelligently calls our APIs using the OpenAPI Specifications. We will also cover strategies for selling our custom GPTs in the GPT stores.

Advanced Specializations

Students will have the option of selecting one of the following specialisations in their fourth quarter:

1. **Web3, Blockchain, and GenAI Integration Specialization**
2. **Metaverse, 3D, and GenAI Integration Specialization**
3. **Healthcare and Medical GenAI Specialization**
4. **GenAI for Accounting, Finance, and Banking Specialization**
5. **GenAI for Engineers Specialization**
6. **GenAI for Sales and Marketing Specialization**
7. **GenAI for Automation and Internet of Things (IoT) Specialisation**
8. **GenAI for Cyber Security**

Common Questions (FAQs) with Detailed Answers

1. **What is Cloud Applied Generative AI Engineering?**
   Cloud Applied Generative AI Engineering (GenEng) is the application of generative AI technologies to solve real-world problems in the cloud.
   - Generative AI is a type of artificial intelligence that can create new data or content from existing data.
   - Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud").

   By combining generative AI with cloud computing, businesses can solve a variety of problems, such as:
   - Creating personalised experiences for customers
   - Automating tasks
   - Improving decision-making
   - Detecting fraud
   - Developing new products and services

   The potential applications of cloud-applied generative AI are endless. As generative AI and cloud computing continue to develop, we can expect to see even more innovative and groundbreaking uses for this technology.

2. **How valuable are the Cloud Applied Generative AI developers?**
   Developers with expertise in Cloud Applied Generative AI were in extremely high demand due to the increasing adoption of GenAI technologies across various industries. However, the supply of developers skilled specifically in this niche area might not have been as abundant compared to more generalised AI or cloud computing roles.

   The demand for AI developers, especially those proficient in applying generative AI techniques within cloud environments, has been rising due to the growing interest in using AI for creative applications, content generation, image synthesis, natural language processing, and other innovative purposes.

   According to some sources, the average salary for a Cloud Applied Generative AI developer in the global market is around $150,000 per year. However, this may vary depending on the experience level, industry, location, and skills of the developer. For example, a senior Cloud Applied Generative AI developer with more than five years of experience can earn up to $200,000 per year. A Cloud Applied Generative AI developer working in the financial services industry can earn more than a developer working in the

entertainment industry. A Cloud Applied Generative AI developer working in New York City can earn more than a developer working in Dubai. In general, highly skilled AI developers, especially those specialising in applied generative AI within cloud environments, tend to earn competitive salaries that are often above the average for software developers or AI engineers due to the specialised nature of their skills. Moreover, as generative AI technology becomes more widely adopted and integrated into various products and services, the demand for Cloud Applied Generative AI developers is likely to increase.

Therefore, Cloud Applied Generative AI developers are valuable professionals who have a bright future ahead of them. They can leverage their creativity and technical skills to create innovative solutions that can benefit various industries and domains. They can also enjoy very competitive salary and career growth opportunities.

3. **What is the potential for Cloud Applied Generative AI Developers to start their own companies?**
   Cloud Applied Generative AI Developers have a significant potential to start their own companies due to several factors:

1. **Emerging Field:** Generative AI, particularly when applied within cloud environments, is still an evolving field with immense potential for innovation. Developers who understand the intricacies of both generative AI techniques and cloud technologies can identify unique opportunities to create novel products, services, or solutions.
2. **Market Demand:** There is a growing demand for AI-driven applications, especially those that involve generative capabilities such as image generation, content creation, style transfer, etc. Developers with expertise in this area can leverage this demand to create specialized products that cater to specific industries or consumer needs.
3. **Innovation and Differentiation:** The ability to develop unique and innovative solutions using generative AI in the cloud can set apart these developers' startups from more conventional companies. They can explore new ways of generating content, enhancing user experiences, or solving complex problems with AI-generated solutions.
4. **Access to Cloud Resources:** Cloud platforms provide scalable and cost-effective resources that are crucial for AI development. Developers starting their own companies can leverage cloud services to access powerful computing resources, storage, and AI-related services without significant upfront investment.
5. **Entrepreneurial Opportunities:** Developers with entrepreneurial spirit and a deep understanding of AI technologies can identify gaps in the market and

build startups to fill those gaps. They can create platforms, tools, or services that simplify the adoption of generative AI for businesses or developers.

6. **Collaboration and Partnerships:** These developers can collaborate with other experts in AI, domain specialists, or businesses to create innovative solutions or explore new application areas for generative AI in the cloud.

   However, starting a company, especially in a specialized field like Cloud Applied Generative AI, requires more than technical expertise. It also demands business acumen, understanding market needs, networking, securing funding, managing resources effectively, and navigating legal and regulatory landscapes.

   Successful entrepreneurship in this domain involves a combination of technical skills, innovation, a deep understanding of market dynamics, and the ability to transform technical expertise into viable products or services that address real-world challenges or opportunities.

   Developers aspiring to start their own companies in the Cloud Applied Generative AI space can do so by conducting thorough market research, networking with industry experts, building a strong team, and developing a clear business plan that highlights the unique value proposition of their offerings.

   To sum up, the potential for Cloud Applied Generative AI Developers to start their own companies is high.

   - Generative AI is a rapidly growing field with a high demand for skilled professionals.
   - The Certified Generative AI (GenEng) Developer and Engineering Program provides students with the skills and knowledge they need to develop and apply cutting-edge generative AI technologies.
   - The program also teaches students how to start and run a successful business.
   - Graduates of the program will be well-positioned to start their own companies and capitalise on the growing demand for generative AI solutions.

4. **Why do we have to learn two programming languages?**
   You are learning two programming languages in the first quarter of the GenEng certification program because they are both essential for developing smart applications with GenAI.

   - **TypeScript (Which is a superset of JavaScript) is used for building user interfaces**, and it is a relatively new programming language that

is gaining popularity due to its strong typing system and its ability to be used with JavaScript, React, and Next.js.
- **Python is used for developing application programming interfaces (APIs)**, and it is a more established programming language that is known for its versatility and ease of use. It is also the go-to language for developing AI systems.

5. **Is the program not too long, one year is a long time?**
   The length of the program is one year which is broken down into four quarters of three months each. The program covers a wide range of topics including TypeScript, Python, Front-end Development, GenAI, API, Database, Cloud Development, and DevOps. The program is designed to give students a comprehensive understanding of generative AI and prepare them for careers in this field. Nothing valuable can be achieved overnight, there are no shortcuts in life.

6. **Why don't we use TypeScript (Node.js) to develop APIs in the third quarter instead of using Python, this way we don't have to learn two programming languages?**
   The Certified Generative AI (GenEng) Developer and Engineering Program teaches students to develop smart applications using both TypeScript and Python. We will not use Typescript in GenAI API development because Python is a priority with the AI community when working with AI and if any updates come in libraries they will first come for Python. Python is always a better choice when dealing with AI and API.

   - **Python is the de facto standard for AI Development**.
   - TypeScript is a more modern language that is gaining popularity for Web Development, but Python is more widely used and has a larger ecosystem of libraries and frameworks available, especially for AI.
   - TypeScript is used for web user interfaces, while Python is used for APIs.
   - In the third quarter, students will learn to develop APIs using Python instead of TypeScript.
   - Python is a more commonly used language for AI and API development, and it has a larger ecosystem of libraries and frameworks available for these purposes.
   - TypeScript is a more modern language that is becoming increasingly popular for API development also, but it is still not as widely used as Python, especially for AI applications and development.
   - By teaching students both TypeScript and Python, the program prepares them to work with a variety of technologies and solve a wider range of problems.

7. **What is the difference between OpenAI Completion API, OpenAI Assistant API, Google Gemini Multi-Modal API, and LangChain?**
   The difference between OpenAI Completion API, OpenAI Assistant API, Google Gemini Multi-Modal API, and LangChain is that they are different ways of using artificial intelligence to generate text, images, audio, and video based on some input, but they have different features and applications. Here is a summary of each one:

   OpenAI Completion API is the most fundamental OpenAI model that provides a simple interface that's extremely flexible and powerful. You give it a prompt and it returns a text completion, generated according to your instructions. You can think of it as a very advanced autocomplete where the language model processes your text prompt and tries to predict what's most likely to come next. The Completion API can be used for various tasks such as writing stories, poems, essays, code, lyrics, etc. It also supports different models with different levels of power suitable for different tasks.

   OpenAI Assistant API is an interface to OpenAI's most capable model (gpt-4) and their most cost-effective model (gpt-3.5-turbo). It provides a simple way to take text as input and use a model like gpt-4 or gpt-3.5-turbo to generate an output. The Assistant API allows you to build AI assistants within your applications. An Assistant has instructions and can leverage models, tools, and knowledge to respond to user queries. The Assistant API currently supports three types of tools: Code Interpreter, Retrieval, and Function calling.

   Google Gemini Multi-Modal API is a new series of foundational models built and introduced by Google. It is built with a focus on multimodality from the ground up. This makes the Gemini models powerful against different combinations of information types including text, images, audio, and video. Currently, the API supports images and text. Gemini has proven by reaching state-of-the-art performance on the benchmarks and even beating the ChatGPT and the GPT4-Vision models in many of the tests.
   There are three different Gemini models based on their size, the Gemini Ultra, Gemini Pro, and Gemini Nano in decreasing order of their size.

   LangChain is a platform that allows you to interact with various language models from different providers such as OpenAI, Google Gemini, Hugging Face Transformers, etc. You can use LangChain to create applications that leverage the power of natural language processing without having to deal with the complexity of APIs or SDKs. LangChain provides a user-friendly interface that lets you choose the model you want to use, customize the parameters you want to apply, and see the results in real-time.

8. **Why don't we use Flask or Django for API development instead of FastAPI?**
    - **FastAPI is a newer and more modern framework than Flask or Django.** It is designed to be fast, efficient, and easy to use. FastAPI is also more scalable than Flask or Django, making it a better choice for large-scale projects.
    - **FastAPI is also more feature-rich than Flask or Django.** It includes several built-in features that make it easy to develop APIs, such as routing, validation, and documentation.
    - **Overall, FastAPI is a better choice for API development than Flask or Django.** It is faster, more scalable, and more feature-rich.

9. **Why do we need to learn Cloud technologies in a Generative AI program?**
    Cloud technologies are essential for developing and deploying generative AI applications because they provide a scalable and reliable platform for hosting and managing complex workloads.

    - Cloud computing offers a vast pool of resources that can be provisioned on demand, which is ideal for generative AI applications that can be computationally intensive.
    - Cloud providers offer a wide range of services that can be used to support generative AI applications, including storage, computing, networking, and machine learning.
    - Cloud services are typically more cost-effective than on-premises infrastructure, which can be a significant advantage for generative AI applications that are often used for large-scale projects.

    The Certified Generative AI (GenEng) Developer and Engineering Program teaches you how to use a variety of cloud services, including Google Cloud Run, Azure Container Apps, and Kubernetes, to deploy your applications to the cloud. You will also learn how to use **Docker containers** to package and deploy your applications, and how to use Terraform to manage your cloud infrastructure.

    By the end of the program, you will be able to:
    - Use Docker containers to package and deploy your applications
    - Develop and deploy generative AI applications to the cloud
    - Manage your cloud infrastructure using Terraform

10. **What is the purpose of Docker Containers and what are the benefits of deploying them with Docker Compose, Google Cloud Run, Azure Container Apps, and Kubernetes?**
    - **Docker Containers** are a way to package software into a single unit that can be run on any machine, regardless of its operating system. It is used to create a Dockerfile, which is a text file that describes how to build a Docker image. The image is then used to create a container, which is a running instance of the image. This makes them ideal for deploying applications on a variety of platforms, including cloud-based services.
    - **Docker Compose** is a tool provided by Docker that allows you to define and manage multi-container Docker applications locally. It enables you to use a YAML file to configure the services, networks, and volumes needed for your application's setup. With Docker Compose, you can describe the services your application requires, their configurations, dependencies, and how they should interact with each other, all in a single file. This makes it easier to orchestrate complex applications locally composed of multiple interconnected containers.
    - **Google Cloud Run** is a serverless computing platform that allows you to run stateless containers that are invocable via HTTP requests. It is fully managed, so you don't need to worry about provisioning or managing servers.
    - **Azure Container Apps** is a serverless platform from Microsoft that allows you to maintain less infrastructure and save costs while running containerized applications. Instead of worrying about server configuration, container orchestration, and deployment details, Container Apps provides all the up-to-date server resources required to keep your applications stable and secure.
    - **Kubernetes** is a container orchestration system that automates the deployment, scaling, and management of containerized applications. It allows you to run multiple containers on a single machine or across multiple machines. It is an open source and can be deployed in your data center or the cloud.

11. **Why do we need to learn Web development technologies in a Generative AI program?**
    Web development technologies are essential for developing and deploying generative AI applications because they allow you to **create user interfaces** that allow users to interact with your applications.
    The Certified Generative AI (GenEng) Developer and Engineering Program teaches you how to use cutting-edge web development technologies, including TypeScript, React, Next.js, and Tailwind CSS, to build and deploy state-of-the-art web user interfaces. You will also learn how to use Vercel AI SDK, an open-source library for building AI-powered user interfaces.

12. **What is the purpose of learning to develop APIs in a Generative AI program?**
    APIs (Application Programming Interfaces) are used to connect different software applications and services together. They are the building blocks of the internet and are essential for the exchange of data between different systems.

    In the third quarter of the Certified Generative AI (GenEng) Developer and Engineering Program, students will learn to develop APIs not just as a backend for their front end but also as a **product** itself. In this model, the API is at the core of the business's value.

    - APIs are used to make it possible for different software applications to communicate with each other.
    - APIs are used to access data from a remote server.
    - APIs are used to create new services or applications that are integrated with existing systems.
    - APIs are used to improve the security of applications by providing a way to control access to data.
    - By learning to develop APIs, students will gain the skills necessary to create powerful and efficient software applications that can be used to solve a variety of business problems.

13. **What is the purpose of using Python-based FastAPI and related technologies in Quarter 3?**
    In the third quarter of the Certified Generative AI (GenEng) Developer and Engineering Program, students will learn how to use Python-based FastAPI as a core library for API development.

    - FastAPI is a high-performance, lightweight, and easy-to-use framework for building APIs.

- It is designed to be fast, scalable, and secure.
- FastAPI is compatible with a wide range of programming languages and frameworks, making it a good choice for developers with different skill sets.
- Students will also learn about the following related technologies:
- **Pedantic:** Pedantic is a Python library that helps to improve the quality of your code by checking for errors and potential problems.
- **SQLAlchemy:** SQLAlchemy is a Python library that provides an object-relational mapping (ORM) layer for working with databases.
- **PostgreSQL:** PostgreSQL is a free and open-source relational database management system (RDBMS) that can be used for development. Highly scalable database systems compatible with it have also been deployed by all the major cloud platforms.

By the end of the quarter, students will be able to use Python-based FastAPI to develop APIs that are fast, scalable, and secure.

14. What does the API-as-a-Product model entail?

API-as-a-Product is a type of Software-as-a-Service that monetizes niche functionality, typically served over HTTP. In this model, the API is at the core of the business's value. The API-as-a-Product model is different from the traditional API model, where APIs are used as a means to access data or functionality from another application. In the API-as-a-Product model, the API itself is the product that is being sold.

The benefits of the API-as-a-Product model include:

- **Increased flexibility:** APIs can be used to access data or functionality from any application, regardless of the underlying platform or technology. This gives businesses greater flexibility in how they integrate APIs into their applications.
- **Reduced development costs:** APIs can be reused by multiple applications, which can save businesses the time and expense of developing their custom APIs.
- **Improved scalability:** APIs can be scaled up or down as needed, which makes them well-suited for businesses with fluctuating or unpredictable traffic demands.
- **Enhanced security:** APIs can be more secure than traditional methods of data exchange, as they can be protected by a variety of security measures, such as encryption and access control.

15. **What are the benefits of using Docker Containers for development, testing, and deployment?**
    Docker Containers are a fundamental building block for development, testing, and deployment because they provide a consistent environment that can be used across different systems. This eliminates the need to worry about dependencies or compatibility issues, and it can help to improve the efficiency of the development process. Additionally, Docker Containers can be used to isolate applications, which can help to improve security and make it easier to manage deployments.

16. **Why in this program are we not learning to build LLMs ourselves? How difficult is it to develop an LLM like ChatGPT 4 or Google's Gemini?**
    Developing an LLM like ChatGPT 4 or Google Gemini is extremely difficult and requires a complex combination of resources, expertise, and infrastructure. Here's a breakdown of the key challenges:

    Technical hurdles:

    Massive data requirements: Training these models requires an immense amount of high-quality data, often exceeding petabytes. Compiling, cleaning, and structuring this data is a monumental task.
    Computational power: Training LLMs demands incredible computational resources, like high-performance GPUs and specialised AI hardware. Access to these resources and the ability to optimise training processes are crucial.
    Model architecture: Designing the LLM's architecture involves complex decisions about parameters, layers, and attention mechanisms. Optimising this architecture for performance and efficiency is critical.
    Evaluation and bias: Evaluating the performance of LLMs involves diverse benchmarks and careful monitoring for biases and harmful outputs. Mitigating these biases is an ongoing research challenge.

    Resource and expertise:

    Team effort: Developing an LLM like ChatGPT 4 or Google Gemini requires a large team of experts across various disciplines, including AI researchers, machine learning engineers, data scientists, and software developers.
    Financial investment: The financial resources needed are substantial, covering costs for data acquisition, hardware, software, and talent. Access to sustained funding is critical.

Additionally:

Ethical considerations: LLMs raise ethical concerns like potential misuse, misinformation, and societal impacts. Responsible development and deployment are crucial.
Rapidly evolving field: The LLM landscape is constantly evolving, with new research, models, and benchmarks emerging. Staying abreast of these advancements is essential.

Therefore, while ChatGPT 4 and Google Gemini have made impressive strides, developing similar LLMs remains a daunting task accessible only to a handful of organizations with the necessary resources and expertise.

In simpler terms, it's like building a skyscraper of knowledge and intelligence. You need the right materials (data), the right tools (hardware and software), the right architects (experts), and a lot of hard work and attention to detail to make it stand tall and function flawlessly.

Developing similar models would be a daunting task for individual developers or smaller teams due to the enormous scale of resources and expertise needed. However, as technology progresses and research findings become more accessible, it might become incrementally more feasible for a broader range of organizations or researchers to work on similar models, albeit at a smaller scale or with fewer resources. At that time we might also start to focus on developing LLMs ourselves.

To sum up, the focus of the program is not on LLM model development but on applied Cloud GenAI Engineering (GenEng), application development, and fine-tuning of foundational models. The program covers a wide range of topics including TypeScript, Python, Front-end Development, GenAI, API, Database, Cloud Development, and DevOps, which will give students a comprehensive understanding of generative AI and prepare them for careers in this field.

17. **Business wise does it make more sense to develop LLMs ourselves from scratch or use LLMs developed by others and build applications using these tools by using APIs and/or fine-tuning them?**
Whether it makes more business sense to develop LLMs from scratch or leverage existing ones through APIs and fine-tuning depends on several factors specific to your situation. Here's a breakdown of the pros and cons to help you decide:

Developing LLMs from scratch:

Pros:

Customization: You can tailor the LLM to your specific needs and data, potentially achieving higher performance on relevant tasks.
Intellectual property: Owning the LLM allows you to claim intellectual property rights and potentially monetize it through licensing or other means.
Control: You have full control over the training data, algorithms, and biases, ensuring alignment with your ethical and business values.

Cons:

High cost: Building and training LLMs require significant technical expertise, computational resources, and data, translating to high financial investment.
Time commitment: Developing an LLM is a time-consuming process, potentially delaying your go-to-market with your application.
Technical expertise: You need a team of highly skilled AI specialists to design, train, and maintain the LLM.

Using existing LLMs:

Pros:

Lower cost: Leveraging existing LLMs through APIs or fine-tuning is significantly cheaper than building them from scratch.
Faster time to market: You can quickly integrate existing LLMs into your applications, accelerating your launch timeline.
Reduced technical burden: You don't need a large team of AI specialists to maintain the LLM itself

Cons:

Less customization: Existing LLMs are not specifically designed for your needs, potentially leading to lower performance on some tasks.
Limited control: You rely on the data and biases of the existing LLM, which might not align with your specific requirements.
Dependency on external parties: You are dependent on the availability and maintenance of the LLM by its developers.

Here are some additional factors to consider:

The complexity of your application: Simpler applications might benefit more from existing LLMs, while highly complex ones might require the customization of a dedicated LLM.
Your available resources: If you have the financial and technical resources, developing your own LLM might be feasible. Otherwise, existing options might be more practical.
Your competitive landscape: If your competitors are using LLMs, you might need to follow suit to remain competitive.
Ultimately, the best decision depends on your specific needs, resources, and business goals. Carefully evaluating the pros and cons of each approach will help you choose the strategy that best aligns with your success.

18. **What are Custom GPTs?**

"Custom GPTs" refers to specialised versions of the Generative Pre-trained Transformer (GPT) models that are tailored for specific tasks, industries, or data types. These custom models are adapted from the base GPT architecture, which is a type of language model developed by OpenAI. Custom GPT models are trained or fine-tuned on specific datasets or for particular applications, allowing them to perform better in those contexts compared to the general-purpose models.

Here are some examples of what custom GPT models might be used for:

1. **Industry-Specific Needs**: A custom GPT for legal, medical, or financial industries could be trained on domain-specific texts to understand and generate industry-specific language more accurately.

2. **Language and Localization**: Models can be customised for different languages or dialects that might not be well-represented in the training data of the base model.

3. **Company-Specific Applications**: Organisations might develop a custom GPT model trained on their own documents and communications to assist with internal tasks like drafting emails, generating reports, or providing customer support.

4. **Educational Purposes**: Educational institutions might develop custom GPTs trained on educational material to assist in creating teaching materials or providing tutoring in specific subjects.

5. **Creative Writing and Entertainment**: Custom models could be trained on specific genres of literature or scripts to assist in creative writing or content creation.

6. **Technical and Scientific Research**: A custom GPT model could be trained on scientific literature to assist researchers in summarising papers, generating hypotheses, or even drafting new research.

These custom models are created through a process of fine-tuning, where the base GPT model is further trained (or 'fine-tuned') on a specific dataset. This process allows the model to become more adept at understanding and generating text that is relevant to the specific use case. Fine-tuning requires expertise in machine learning and natural language processing, as well as access to relevant training data.

19. **What are Actions in GPTs?**
    Actions are a way to connect custom GPTs to external APIs, allowing them to access data or interact with the real-world. For example, you can use actions to create a GPT that can book flights, send emails, or order pizza. **Actions are defined using the OpenAPI specification**, which is a standard for describing APIs. You can import an existing OpenAPI specification or create a new one using the GPT editor.

20. **What are the different specialisations offered at the end of the program and what are their benefits?**
    At the end of the GenEng certification program we offer six specialisations in different fields:

    Web3, Blockchain, and GenAI Integration: This specialisation will teach students how to integrate generative AI with Web3 and blockchain technologies. This is relevant to fields such as finance, healthcare, and supply chain management.
    Benefits:
    - Learn how to create smart contracts and decentralised applications (dApps).
    - Gain a deeper understanding of the potential of blockchain technology and how it can be used to improve business processes.
    - Develop the skills necessary to work in a rapidly growing field with high demand for skilled professionals.

    Metaverse, 3D, and GenAI Integration: This specialisation will teach students how to create and use 3D models and other immersive content manually and with generative AI. This is relevant to fields such as gaming, marketing, and architecture.

Benefits:

- Learn how to use generative AI to create realistic and immersive 3D models.
- Develop the skills necessary to work in the growing field of virtual reality (VR) and augmented reality (AR).
- Apply generative AI to solve real-world problems in areas such as product design, marketing, and education.

Healthcare and Medical GenAI: This specialization will teach students how to use generative AI to improve healthcare and medical research. This is relevant to fields such as drug discovery, personalized medicine, and surgery planning.
Benefits:

- Learn how to use generative AI to identify diseases, develop new drugs, and personalize treatment plans.
- Gain a deeper understanding of the ethical implications of using generative AI in healthcare.
- Prepare for a career in a growing field with high demand for skilled professionals.

GenAI for Accounting, Finance, and Banking: This specialisation will teach students how to use generative AI to improve accounting, finance, and banking processes. This is relevant to fields such as fraud detection, risk management, and investment analysis.
Benefits:

- Learn how to use generative AI to automate tasks, identify patterns, and make predictions.
- Gain a deeper understanding of the financial industry and how generative AI can be used to improve its processes.
- Prepare for a career in a growing field with high demand for skilled professionals.

GenAI for Engineers: This specialisation will teach students how to use generative AI to improve engineering design and problem-solving. This is relevant to fields such as manufacturing, construction, and product development.
Benefits:

- Learn how to use generative AI to create simulations, optimize designs, and predict failures.
- Gain a deeper understanding of the engineering design process and how generative AI can be used to improve it.
- Prepare for a career in a growing field with high demand for skilled professionals.

GenAI for Sales and Marketing: This specialisation will teach students how to use generative AI to improve sales and marketing campaigns. This is relevant to fields such as advertising, public relations, and customer service. Benefits:

- Learn how to use generative AI to create personalised marketing messages, generate leads, and track campaign performance.
- Gain a deeper understanding of the latest marketing trends and how generative AI can be used to improve them.
- Prepare for a career in a growing field with high demand for skilled professionals.

GenAI for Automation and Internet of Things (IoT):

- **Provide Multi-Modal User Interface for the IoT systems:** Multimodal interaction exploits the synergic use of different modalities to optimise the interactive tasks accomplished by the users. This allows a user to use several input modes such as speech, touch, and visual to interact with IoT systems.
- **Improve efficiency and accuracy of industrial processes:** By implementing GenAI in automation and IoT systems, industries can optimise their processes, reduce manual labour, and increase productivity while ensuring higher accuracy and consistency.
- **Enhance decision-making:** GenAI can analyse vast amounts of data collected by IoT sensors to derive valuable insights, enabling businesses to make informed decisions regarding operations, maintenance, and resource allocation.
- **Personalise user experiences:** GenAI can leverage IoT data to understand user preferences and behaviours, enabling the creation of personalised experiences across smart devices and IoT-enabled systems.

GenAI for Cyber Security:

- **Strengthen threat detection and response:** GenAI can be used to rapidly detect and respond to cyber threats by analysing large volumes of security data in real time, identifying anomalies, and suggesting appropriate countermeasures.
- **Enhance security monitoring and analysis:** GenAI can assist security analysts in monitoring and analysing security logs, automating threat detection, and providing insights into security risks and vulnerabilities.
- **Improve threat intelligence:** GenAI can be used to gather and analyze threat intelligence from various sources, enabling organisations to stay informed about the latest threats and trends and proactively strengthen their security posture.