Project 1: FactGuard: Leveraging Large Language Models for Video Content Factual Verification

	Content
Project Title	FactGuard: Leveraging Large Language Models for Video Content Factual Verification
Category	Generative AI
Problem Statement	This project aims to develop a content verification system leveraging Large Language Models (LLMs) to enhance the quality and reliability of video content, particularly on online platforms like YouTube. The system will employ Automatic Speech Recognition (ASR) technology to transcribe video content into text, which will then be compared against existing web pages for accuracy assessment. LLMs, specifically GPT models, will be utilized for semantic understanding and fine-grained analysis of the textual content, enabling the detection of inaccuracies or discrepancies. Refer - Youtube, Wikipedia
Tech Stack	LLM, MERN
Max Groups	1

Project 2: Mental-LLM: Predicting Mental Health from Online Text with LLM

	Content
Project Title	Mental-LLM: Predicting Mental Health from Online Text with LLM
Category	Generative AI
Problem Statement	Mental-LLM presents a novel methodology for leveraging Large Language Models (LLMs) in mental health prediction tasks using given data. This research explores the effectiveness of advanced language models in analyzing online textual content to predict mental health conditions. Furthermore, this study conducts a comparative analysis of multiple LLMs, including Alpaca, Alpaca-LoRA, FLAN-T5, to determine the optimal model for mental health prediction. This research contributes to advancing mental health assessment methodologies by providing insights into the capabilities of LLMs and guiding the selection of the most suitable model for mental health prediction tasks. Reflinks: https://github.com/neuhai/Mental-LLM https://arxiv.org/abs/2307.14385 Student Mental health - https://www.kaggle.com/datasets/mariaren/covid19-healthy-diet-dataset Fetal Health Classification - https://www.kaggle.com/datasets/csafrit2/maternal-health-risk-data
Tech Stack	LLM, MERN
Max Groups	1

Project 3: Automated Vehicle Damage Assessment : Advancements in Visual-Language Models

	Content
Project Title	Automated Vehicle Damage Assessment : Advancements in Visual-Language Models
Category	Generative AI
Problem Statement	Proposed a novel approach leveraging the Contrastive Language-Image Pretraining (CLIP) model / for automated vehicle damage assessment. CLIP, developed by OpenAI, is a powerful visual-language model capable of understanding the relationship between images and text through contrastive learning. We fine-tune the CLIP model on a curated dataset comprising images of damaged vehicles paired with corresponding textual descriptions of the damage. The fine-tuned CLIP model learns to generate descriptive reports for vehicle damage solely based on input images, enabling automated assessment without manual intervention.
Tech Stack	VLM, MERN
Max Groups	1

Project 4: Intellijobs:Automating Job Portal Scraping and Analysis with Large Language Models

	Content
Project Title	Intellijobs:Automating Job Portal Scraping and Analysis with Large Language Models
Category	Generative AI
Problem Statement	The aim of this project is to develop a system that automates the process of scraping job listings from a selected job portal and analyzing them using Large Language Models (LLMs) to provide valuable insights to job seekers and employers. The system will leverage web scraping techniques to collect job listings (like data analyst) and natural language processing capabilities of LLMs to analyze job descriptions like highest salary, job title, location, company ratings, company_size,employment_type and to extract anyother key information.

Choose a Job Portal: Select a job portal that you'll be scraping for job listings. Popular choices might include Indeed, LinkedIn, Glassdoor, or specific industry-focused job boards.
Web Scraping: Develop a web scraping script using libraries like BeautifulSoup (for HTML parsing) or Scrapy (for more advanced web scraping tasks) in Python. This script will fetch job listings from the selected portal, extracting relevant information such as job title, company, location, job description, requirements, and any other details deemed important.
Data Cleaning and Preprocessing: Clean and preprocess the scraped data to remove noise, standardize text formats, handle missing values, and perform any necessary transformations. The data obtained from the data cleaning should be stored in required format.
Train a custom LLM using the preprocessed job description data. You can use frameworks like Hugging Face's Transformers or OpenAI's GPT (if you have access) for training. Fine-tune a pre-trained LLM (e.g., GPT-2, BERT) on your job description dataset using transfer learning techniques.
User Interface (UI):
Develop a user-friendly interface where users can input their criteria (e.g., location, job title, skills) and receive relevant insights. |
| Tech Stack | LLM,MERN |

	Content
Max Groups	1

Project 5: Chat bot for Medical Queries

	Content
Project Title	Chat bot for Medical Queries
Category	Generative AI
Problem Statement	LLMs have numerous applications in healthcare domain. In scenarios where doctors may not be easily available, Gen AI models will be useful for addressing patient queries, at least for non-critical diseases. A pre-trained LLM model is a good candidate for developing a Medical Chat Doctor. It should be able to answer patient queries. To provide relevant answers and improve the quality of diagnosis, the model should be fine tuned with custom data. Choose an appropriate model like (Mistral-7B) and search for training data set for fine tuning the model. The patient should be able input problems/symptoms he/she is facing and ask related questions. An appropriate response should be given by the Chat-bot in a couple of sentences. https://huggingface.co/datasets/lavita/ChatDoctor-HealthCareMagic-100K

Content
Acquire Dataset: Obtain a dataset containing medical questions, symptoms, diagnoses, treatments, and related information. Consider using publicly available datasets, healthcare forums, or electronic health records (EHR) with appropriate permissions. Preprocess Data: Clean and preprocess the medical dataset to remove noise, standardize text formats, handle abbreviations, and perform any necessary transformations. Ensure that the dataset is suitable for fine-tuning the pre-trained model. Fine-tune Model: Fine-tune with LLM like Mistral-7B with the preprocessed medical dataset. Utilize transfer learning techniques to update the model's parameters based on the medical data, adapting it to the task of answering patient queries and providing relevant medical information. Develop User Interface: Design and develop a user-friendly interface for the Medical Chat Doctor application. Allow patients to input their symptoms, ask related questions, and receive appropriate responses from the chatbot.

Tech Stack	LLM,MERN
Max Groups	1

Project 6: Chat with Author

	Content
Project Title	Chat with Author
Category	Generative AI
Problem Statement	An Agent Helps to Automate chat with Author while Providing the Query provides accuracy results.
	we'll create an application that enables you to ask questions about Selected Authors Textbook(PDFs) and receive accurate answers.
	Train GPT on PDF documents and fine-tune it to your specific use case.
	Experience the seamless user interface as you upload PDFs, ask questions, and receive prompt answers from the LLM.
	Prerequisites: (an open-source Python,Javascript, Flask framework, Basics on OpenAI) Discover Langchain's capabilities in training GPT models on your data and generating personalized LLMs.
	Explore text embeddings and their integration with Langchain using OpenAI's API. Automate tasks and improve efficiency using Langchain with Streamlit. Outcome: You'll get Q&A on given topic You'll learn how to use LangChain (a framework that makes it easier to assemble the components to build a chatbot) & Pinecone - a 'vectorstore' to store your documents in number 'vectors'. You'll also learn how to create a frontend chat interface to display the results alongside source document
Tech Stack	LLM,MERN
Max Groups	1

Project 7: Retail LLM

	Content
Project Title	Retail LLM
Category	Generative AI
Problem Statement	"leveraging LLM technology in retail operations can lead to a more seamless and personalized shopping experience for customers, empower salespersons with actionable insights, and provide mart owners with valuable tools for financial analysis and decision-making across their branches 1. Customer Experience Enhancement Improved Search Capabilities: LLM-powered search functionalities can understand natural language queries, enabling customers to find products more intuitively and efficiently. Personalized Product Recommendations: By analyzing customer browsing and purchasing history, LLMs can provide tailored product recommendations, enhancing the likelihood of conversion and customer satisfaction. 2. Salesperson Support Product Availability and Sales Trends: LLMs can analyze sales data across various branches or locations in real-time, providing insights into product availability and identifying trends where certain products may be underperforming. Salespersons can leverage this information to adjust their sales strategies accordingly. Sales Performance Analysis: LLMs can analyze sales data to identify patterns and factors influencing

Content
the progression of certain product sales. This analysis can help salespersons understand customer preferences and adjust their approach to promote these products effectively. 3. Financial Analysis for Mart Owners Branch Performance Summary: LLMs can consolidate data from multiple branches to provide mart owners with a comprehensive summary of financial performance. This includes metrics such as sales revenue, profit margins, and inventory turnover rates, allowing owners to identify areas for improvement and make informed business decisions."

Tech Stack	LLM,MERN
Max Groups	1

Project 8: Cricket LLM

	Content
Project Title	Cricket LLM
Category	Generative AI
Problem Statement	Cricket is played in various formats. A specialized Cricket Large Language Model (LLM) can be used for match summaries and reports, player profiling and analysis, cricket news aggregation, historical data analysis and many more such acivities. The aim of this project is to create a Cricket LLM. It would be built - primarily - using web scraping. Here's a general outline of how you could approach building such a system: Define Requirements: Determine the specific data that would be scraped. Choose the websites to scrape this data from. Tool for scraping: Scrapy would be used for this purpose. Where needed, necessary code in Python would be developed. Automated scraping scripts would be setup to fetch data at regular intervals for live updates. Data Processing and Storage: The scraped data would be processed to extract relevant information and clean it where necessary. It would be put in a structured format (e.g., CSV file) for easy processing. Fine-tuning of the LLM would be done using the above processed data. Build the User Interface: A UI would be developed where users can view live cricket data. This could be a web application, desktop application, mobile app or chatbot, enabling display of live scores, player statistics, match events, and other information of interest in a user-friendly manner. This UI would be tested to ensure a stable user experience.
Tech Stack	LLM,MERN
Max Groups	1

Project 9: LLM-driven Insurance advisor

	Content
Project Title	LLM-driven Insurance advisor
Category	Generative AI
Problem Statement	The aim of this project is to develop a LLM-based system that provides personalized recommendations, information, and guidance to users seeking insurance-related assistance. Leveraging the power of web scraping, this system gathers real-time data from various insurance-related websites, extracting relevant information such as policy details, coverage options and premiums from a selected insurance portal.

Content
Here are the likely steps involved: 1. Choose a specific insurance website as the primary source for scraping insurance-related information. (For example PolicyBazaar.com) 2. Develop a web scraping script using tools like BeautifulSoup or Scrapy to fetch the necessary details. 3. Clean and preprocess the scraped data to remove noise, standardize text formats, handle missing values, and perform any necessary transformations. It would also be stored in required format; for example a CSV file. 4. Fine-tune a LLM using the above data. 5. Develop an interface (preferably chatbot) for users to interact with the LLM. The interface should be user friendly and help user input their criteria and receive relevant insights. 6. The application must be tested to ensure its stable operation.

Tech Stack	LLM,MERN
Max Groups	1

Project 10: Virtual Legal Assistant using LLM

	Content
Project Title	Virtual Legal Assistant using LLM
Category	Generative AI
Problem Statement	Creating a Virtual Legal Assistant (VLA) using a Language Model (LM) like BERT involves several steps. Below are the steps to create a chatbot where users can input legal queries related to sections, persons involved in cases, or specific legal terms, and receive relevant case information along with the option to download case-related documents: 1.Learn about BERT 2.Data Collection: Gather legal documents, case law databases, court records, and other relevant sources of legal information. 3.Preprocess:Organize and preprocess the data to extract case summaries, court decisions, and relevant metadata (e.g., case names, sections, persons involved). 4.Model Training: Fine-tune a pre-trained Language Model (LM) like BERT on the legal dataset. This involves training the LM to understand legal language and context. Train the model to classify and extract relevant information from user queries, such as identifying sections, names of persons involved, or legal terms like murder, kidnapping, etc. 5.Development of Chat Interface: Develop a chatbot interface where users can input their legal queries. Implement natural language processing (NLP) techniques to analyze and understand user queries, including named entity recognition (NER) for identifying legal terms and entities. 6.Query Processing: Process user queries to extract relevant keywords, sections, and names of persons involved. Utilize the trained LM to search the legal database for cases related to the extracted keywords and entities. 7.Case Retrieval: Retrieve relevant case summaries or court decisions based on the user's query. Provide options for users to browse or filter the search results based on specific criteria (e.g., case type, jurisdiction, date). 8.Document Retrieval: Enable users to download case-related documents, such as court judgments, rulings, or case reports. Ensure proper permissions and access controls are implemented to manage document retrieval securely. 9.User Interaction and Feedback: Implement features for user interaction, such as asking follow-up questions, clarifying queries, or providing additional information. Gather user feedback to continuously improve the chatbot's performance and user experience. 10.Testing and Evaluation: Test the chatbot extensively to ensure it accurately understands user queries and retrieves relevant case information. Evaluate the chatbot's performance using metrics such as precision, recall, and user satisfaction.
Tech Stack	LLM, MERN
Max Groups	1

Project 11: Saanjh Sahayak

	Content
Project Title	Saanjh Sahayak
Category	Generative AI
Problem Statement	Saanjh is the home for the elderly. The home needs help in tracking health records of the elderly and predicting any risk for diseases. Create an application that helps the Home to perform a clinical decision support system and disease risk prediction using an existing LLM model.This application should help doctor to give proper treatment . 1.Understand the architecture and capabilities of BERT-based Language Models (LLMs) in natural language processing tasks. 2.Gather health records of the elderly residents, including medical history, vital signs, lab results, medication history, etc. 3.Preprocess the available datasets. 4.Fine-tune LLM with cleaned data using Multi-Task Learning approach. 5.Verify the fine-tuning process for its quality at each step 6.Design and develop an application interface for doctors and healthcare professionals to input and access health records of elderly residents. 7.Integrate the fine-tuned LLM model into the application framework for disease risk prediction and clinical decision support. 8.Test the application against evaluation metrics:in terms of accuracy, precision, recall, F1-score, and other relevant metrics. 9.Analyze the application's performance in predicting disease risks and providing decision support to doctors. 10.Repeat steps to attain better performance
Tech Stack	LLM, MERN
Max Groups	1

Project 12: Auto-summarization of given text book / Chapter

	Content
Project Title	Auto-summarization of given text book / Chapter
Category	Generative AI / General
Problem Statement	This project aims to summarize and prepare questions and answers for the given text/pdf. The input may be an article or textbook or a topic. It displays the summarized content on the screen if possible otherwise, the output will be stored into the selected location and format in local file system or cloud. This project also include a UI to retrieve the generated summary and Q&A for the given topic or text. User allowed to level of summarization in 3 levels (Abstract/Summary/article). UI provide the environment to accept plain text / upload a text or pdf document. UI displays the result on the screen based on the level of summarization. This application is to be built on Spark cluster to make it scalable. refer - Empowering NLP with Spark NLP and T5 Model: Text Summarization and Question Answering - John Snow Labs
Tech Stack	LLM,MERN, APACHE SAPRK

	Content
Max Groups	1

Project 13: LLM Based Sophisticated User Interface for retrieving data from a dataset / database

	Content
Project Title	LLM Based Sophisticated User Interface for retrieving data from a dataset / database
Category	Generative AI

	Content
Problem Statement	This project takes the English instructions / sentances as the input and convert into PySpark and SQL codes. The result of the generated query can be retrieved and displayed. This application can plot the graphs based on the given English sentence. Examples: auto_df = spark_ai.create_df("2022 USA national auto sales by brand") auto_df.ai.plot("pie chart for US sales market shares, show the top 5 brands and the sum of others") auto_top_growth_df=auto_df.ai.transform("top brand with the highest growth") auto_top_growth_df.show() LLM involved in converting the given English sentence into PySpark/SQL query User interface can be developed using any front-end technology. This can be improved as voice-based queries in future. Refer - Introducing English as the New Programming Language for Apache Spark | Databricks Blog
Tech Stack	LLM, MERN, APACHE SPARK
Max Groups	1

Project 14: PhishGuard: LLM models for Phishing Detection

	Content
Project Title	PhishGuard: LLM models for Phishing Detection
Category	GenAI / Cybersecurity
Problem Statement	Phishing poses a significant threat to customers and organisations alike, exploiting deceptive tactics such as fraudulent emails, websites and messages. This comes in the form of email phishing, spear phishing, vishing, SMS phishing etc.

The goal of this project is to study the application of LLM for categorising emails and detect phishing. A BERT-based model will be fine-tuned for this purpose. An SMTP server, for example APACHE JAMES, would be used in testing the application.

The project would be taken up in the following steps:
1. Gain knowledge on (BERT-based) LLMs and uses.
2. Download the phishing emails datasets.
3. Process the above downloaded dataset to remove any noisy data etc.
4. Fine tune with LLM with this cleaned data.
5. Verify the fine tuning process for its quality at each step.
6. Develop the SMTP server for sending out malicious emails. (Alternatively, an LLM could also be used to send out phishing emails.)
7. Test the application to test against each metric in terms of accuracy, precision, recall and F1-score.
8. Repeat the steps to attain better performance.

Datasets:
https://www.kaggle.com/datasets/mohamedouledhamed/phishing-templates-gpt

https://www.kaggle.com/datasets/subhajournal/phishingemails |
| Tech Stack | LLM, MERN |
| Max Groups | 1 |

Project 15: ChefGuru: Powered by LLM

	Content
Project Title	ChefGuru: Powered by LLM
Category	NA
Problem Statement	Most often, we get into a situation when we want to cook something delicious, however, we are short on ingredients at our home. It also creates a lot of confusion about what to cook with the available ingredients. Further, we might not have accessibility to more ingredients from the nearby market due to various reasons like non-availability.

In such a situation we keep on searching for recipes that match our ingredients online, but hardly get any results and that too takes a lot of time, and ultimately we return to our kitchen disappointed. There s a need for a system that could actually take ingredients as the input and give us the list of most matching recipes that could help us make delicious food at our homes. |

Content
Our project aims to make a user aware of the various dishes which can be cooked from the available set of ingredients being input by a user. Scraping Module - Scraping the dataset for ingredients, recipes, etc. Training Model - Fetching the data and display the output based on the input. Developing UI - Developing a user-friendly UI to present the recipes based on the user inputs Datasets: https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images https://www.kaggle.com/datasets/kanishk307/6000-indian-food-recipes-dataset https://clickhouse.com/docs/en/getting-started/example-datasets/recipes

Tech Stack	LLM, MERN
Max Groups	1

Project 16: LLMs for Automated Verilog RTL Code Generation

	Content
Project Title	LLMs for Automated Verilog RTL Code Generation
Category	GenAI / Semiconductor design
Problem Statement	Verilog is a hardware description language to model and design digital systems. Generating Verilog code is the first step in automating hardware / chip design. The goal of this project is to fine-tune an LLM with Verilog datasets (collected from GitHub and Verilog textbooks) and test / use it in generation of Verilog code for different chips. An attempt would be made to develop a test bench for testing the syntax of the so generated Verilog code. The project would involve the following steps: 1) Download datasets identified from the sources above. 2) Preprocess the data to identify relevant data. 3) Use the above clean data to fine-tune open-source CodeGen LLM. 4) Use this LLM to generate Verilog code for different types of chips. 5) Develop a test-bench for testing the syntax of the above generated code.
Tech Stack	LLM
Max Groups	2

Project 17: LLMfAMAT: LLM for Advanced Materials

	Content
Project Title	LLMfAMAT: LLM for Advanced Materials
Category	GenAI / Computational Material Science
Problem Statement	Mankind has always been known by advancements in materials; for example from stone age to the silicon age! This significance is too well known if one observes the chips driving the current AI revolution: GPU. Not all materials are made from the traditional metals and alloys nor using the traditional manufacturing techniques. The need of the coming times is engineering (new) materials as required (in contrast to engineering of existing materials) based on need. For example, a bio-degradable material for holding water in a bottle, fuel for automobiles (green hydrogen?) etc The goal of this project is to fine-tune LLM for discovering / designing advanced, custom new materials. A MERN-stack based UI might be necessary to test the above LLM. The steps for realising this project would include: 1) Identifying the LLM to be fine-tuned with materials data. 2) Download datasets and pick the relevant subset to be used for fine-tuning. 3) Fine-tune the LLM. 4) Develop test cases for verifying the LLM. 5) Develop a MERN-stack based UI to implement the above test cases. THIS WOULD BE IN TWO PROJECTS FOCUSED ON DIFFERENT TYPES OF MATERIALS. Refer - The Collection of Database and Dataset Resources in Materials Science: https://github.com/sedaoturak/data-resources-for-materials-science.

	Content
	Millions of new materials discovered with deep learning: https://deepmind.google/discover/blog/millions-of-new-materials-discovered-with-deep-learning/ https://github.com/google-deepmind/materials_discovery
Tech Stack	LLM, MERN
Max Groups	2

Project 18: Prompt Injection attacks in LLMs

	Content
Project Title	Prompt Injection attacks in LLMs
Category	LLM Security

	Content
Problem Statement	LLMs too have vulnerabilities that a hacker could exploit. Prompt Injection Attacks are one such where prompts are generated causing the LLM to reveal what it is supposed to hold secure / private. The goal of this project is to replicate this in a chosen LLM. The steps would involve: 1) Identifying the LLM to simulate the vulnerability. 2) Learn LangChain to communicate with the LLM. 3) Generate static as well as dynamic prompts (using LangChain and a Chatbot / MERN-stack based UI) to entice the LLM to release data that is supposed to be secure / private. CHECK WITH THE MENTOR TO KNOW YOUR ELIGIBILITY FOR THIS PROJECT.
Tech Stack	Python, LLM
Max Groups	1

Project 19: Story Generator In The Style Of Given Stories

	Content
Project Title	Story Generator In The Style Of Given Stories
Category	LLM / Gen AI
Problem Statement	Generate a story based on the given prompt in the given author style. The user has to provide few articles/books/stories and give the prompts for telling a story. The proposed Gen AI model has to understand the given stories and generate a new story with the prompted instructions. The generated story should follow the style of given stories.
Tech Stack	LLM, MERN
Max Groups	1

	Content
Project Title	Story Generator In The Style Of Given Stories
Category	LLM / Gen AI
Problem Statement	Generate a story based on the given prompt in the given author style. The user has to provide few articles/books/stories and give the prompts for telling a story. The proposed Gen AI model has to