COURSE CONTENT

LARGE LANGUAGE MODELS LAB								
II Semester: CSE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
2525884	**Foundation**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	4	2	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 60**				**Total Classes: 60**		
Prerequisites: Basic knowledge of Python programming, Natural Language Processing and machine learning.								

Course Overview:

The large language models lab provides hands-on experience in Natural Language Processing and LLM techniques, including tokenization, tagging, and model building using tools like Python and NLTK. It introduces machine learning and deep learning concepts such as Perceptron, Autoencoders, and sequence models, along with modern approaches like prompt engineering and fine-tuning.

Course Objectives:

1. To understand the basics of NLP and Large Language Models.
2. To implement text processing techniques using Python and NLTK.
3. To develop and fine-tune LLM models using deep learning frameworks.
4. To learn advanced LLM techniques such as prompt engineering and RAG.
5. To build real-world NLP applications with ethical AI practices.

Course Outcomes:

1. To provide hands-on experience using tools and frameworks (e.g., Python with PyTorch or TensorFlow)to implement, train, and fine-tune LLMs.
2. To cover recent advances in LLM research and engineering, such as prompting (zero-shot, few-shot),Retrieval-Augmented Generation (RAG), and parameter-efficient fine-tuning (PEFT) methods like LoRA.
3. To teach methods for systematically evaluating and benchmarking models using various metrics andhuman feedback (RLHF).
4. 4. To discuss the challenges and ethical implications of LLMs, including hallucination, bias, privacy, and safety, and how to mitigate them.
5. To learn how to conceive, design, prototype, and test software applications that leverage LLM-basedservices to solve real-world problems

List of Experiments:

1. Write a program to implement word Tokenizer, Sentence and Paragraph Tokenizers. Check how many words are there in any corpus. Also check how many distinct words are there?

2. Write a python program to eliminate stopwords using nltk

3. Write a python program to perform tokenization by word and sentence using nltk.

4. Write a program to implement Perceptron.

5. Write a Program to implement AND OR gates using Perceptron.

6. Applying the Autoencoder algorithms for encoding the real-world data.

7. Word Analysis.

8. Word Generation.

9. Building Chunker.

10. Building POS Tagger.

11. POS Tagging: Hidden Markov Model.

12. POS Tagging: Viterbi Decoding.

13. Demonstrate the use of Expectation Maximization based clustering algorithm.

ELECTRONIC RESOURCES:

1. https://www.nltk.org/
2. https://docs.pytorch.org/tutorials/beginner/nlp/
3. https://www.tensorflow.org/tutorials/text
4. https://huggingface.co/docs/transformers/index
5. https://www.kaggle.com/learn/natural-language-processing
6. https://spacy.io/usage
7. https://colab.research.google.com/

MATERIALS ONLINE:

1. Course template
2. Tutorial question bank
3. Tech talk and Concept Video topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. E-Learning Readiness Videos (ELRV)