Q1) Architecture of LLM :-

Input embeddings : Converts tokens into continuous vector representations capturing semantic and syntatic information.

Positional encoding : Adds information about tokens position to input embeddings to account for token order.

Encoder : Process input text through multiple layers to create hidden states that maintain text context and meaning.

Self - Attention Mechanism : Computes attention scores to weigh the importance of different tokens in the input sequence.

Feed - forward Neural Network : Applies fully connected layers to each token to capture complex interaction.

Decoder Layers : Enables autoregressive generation by attending to previously generated tokens.

Multi - Head Attention : Performs multiple self - attention operations simultaneouly to capture various relationships in the input.

Layer Normalization : Stabilizes the learning process and improves generalization by normalizing outputs within each layer.

Output Layers : Generates task-specific outputs, like predicting the next token in language modeling.

Q2) BERT model:-

Bert is designed to generate a language model so, only the encoder mechanism is used. Sequence of tokens are fed to the transformer model. These tokens are first embedded into vectors and then processed in the neural network.

The output is a sequence of vectors, each corresponding to an input token, providing contextualized representations.

Masked Language model :

Masking Words : BERT hides about 15% of words in a sentence with a special token.

Guessing Hidden words : BERT guesses the masked words by analyzing the context of neighboring words.

Learning Process : Uses a special layer to guess and convert these guesses into probabilities.

Focus on Masked Words: BERT emphasizes predicting masked words to enhance understanding of context and meaning.

Next Sentence Prediction (NSP):

Sentence Pair Relationship: Trains BERT to understand if two sentences are sequential in a document.

Balanced Training Pairs: 50% pairs are consecutive, 50% are random sentences.

Input Processing: Add special tokens, sentence embeddings, and positional embeddings to prepare data to do NSP.

Prediction Mechanism: Uses the output of CLS token to determine sentence connection probabilities through a classification layer and Soft Max.