Weekly Plan (tentative)

Week 1–2: Introduction to Language Models & N-gram Models

- Lecture 1: Overview of language models; unigram/bigram/trigram; smoothing; perplexity.
  - **Minimum:** Compute bigram probabilities; explain perplexity.
  - **Bonus:** Understand add-k & interpolated backoff smoothing. Find cases when the n-gram model works or fails.
- Lab 1:
  - **Minimum:** Set up code env and access to GPU. Build unigram/bigram with add-k smoothing in python; evaluate perplexity on a corpus.
  - **Bonus:** Implement trigram & interpolation; top-k next-word generation & Hit@3 metrics.
- Group Formation: Students form groups of 4–6.

Week 3–4: Word Embeddings

- Lecture 2: Basic Word Embeddings; CBOW Model; Skip-Gram Model.
  - **Minimum:** Understand basic ideas of word2vec and CBOW/Skip-Gram models.
  - **Bonus:** Negative sampling techniques.
- Lab 2:
  - **Minimum:** Train a tiny word2vec (CBOW & SG) on a toy corpus.
  - **Bonus:** Use SG model to do category expansion.
- Checkpoint: Groups present their concept and prototype for feedback.

Week 5-6: RNN/LSTM Language Models

- Lecture 3: Basic ideas of RNN and LSTM; Context when RNN/LSTM outperform RNN;
  - **Minimum:** When RNN/LSTM is better than n-gram; Walk through one training step at a high level;
  - **Bonus:** problems for RNN and how LSTM solves that (BP/Truncated BP).
- Lab 3:

- **Minimum:** Train an LSTM on WikiText-2 for next-sentence generation.
- **Bonus:** Baseline comparison; ablation on different parameter/architecture settings;

Week 7: Mid-term Project Checkpoint.

- Group Presentation: Project Overview, Dataset, Embedding, Retrieval, and What will you do/finish before the final presentation?

Week 8–9: Transformer & Seq2Seq

- Lecture 4: Attention; Encoder/decoder; Transformer; Benefits over RNNs;
  - **Minimum:** Understand why transformers work better than RNN/LSTM; Basic idea of self-attention; Drawback of transformers.
  - **Bonus:** Understand Encoder/Decoder type transformers and their structure; Understand ideas of WordPiece and GPT-1.
- Lab 4:
  - **Minimum:** Use a tiny pretrained decoder for next-token generation; play with temperature/top-k.
  - **Bonus:** Implementing a GPT-1 type language model and train it to finish text generation.

Week 10–11: BERT & GPT-2

- Lecture 5: Polishing & Testing (Balancing, Debugging, User Feedback).
  - **Minimum:** Masked language model; Pretrained model (BERT); Fine-tuning & prompting;
  - **Bonus:** Causal language mode; GPT-2; PEFT (LoRA).
- Lab 5:
  - **Minimum:** Fine-tune a small BERT for text classification.
  - **Bonus:** Use LoRA to fine-tune a GPT-2 for summarization.

Week 12: Final Presentation

- Groups showcase their language model agents to peers and the instructor, demonstrating

their system's full pipeline: dataset, embeddings, retrieval, fine-tuning, inference; highlighting technical performance and creative design.