Certainly! Below is a more detailed breakdown of the **Deep Learning & Generative AI Course Outline**, including additional subtopics, practical implementations, and learning objectives for each class. This expanded version will provide a clearer understanding of what will be covered in each module and class.

Module 1: Introduction to Deep Learning and AI (4 Classes)

Class 1: Introduction to AI and Machine Learning

- **Learning Objectives:**
  - Understand the difference between AI, ML, and DL.
  - Learn key terminologies and concepts in AI and ML.
  - Explore the historical evolution of AI and its impact on modern technology.
  - Gain familiarity with Generative AI, LLMs, Vector Databases, Hugging Face, and LangChain.
  - Understand the importance of Kaggle for building a portfolio and participating in competitions.
- **Topics:**
  - **Overview of AI, ML, and DL:**
    - Definitions and differences.
    - Real-world applications.
  - **Key Concepts:**
    - Generative AI: What it is and its applications.
    - Large Language Models (LLMs): Introduction to models like GPT, BERT, etc.
    - Vector Databases: Role in AI and ML workflows.
    - Hugging Face: Overview of the platform and its tools.
    - LangChain: Introduction to the framework for building LLM-powered applications.
  - **Kaggle:**
    - Importance of Kaggle for learning and job opportunities.
    - How to participate in Kaggle competitions.
  - **Jobs in AI:**
    - Career paths in Deep Learning, LLMs, and Generative AI.
    - Skills required for AI roles.

Class 2: Basics of Neural Networks

- **Learning Objectives:**
  - Understand the structure and function of artificial neurons.
  - Learn about activation functions and their roles in neural networks.
  - Explore the architecture of Artificial Neural Networks (ANNs).
  - Understand forward and backward propagation.

    - Implement a simple neural network in Python.
- **Topics:**
    - **Artificial Neurons:**
        - Structure and mathematical representation.
    - **Activation Functions:**
        - Linear, Sigmoid, Softmax, Tanh, ReLu, Leaky ReLu.
        - Dying ReLu problem and solutions.
    - **ANN Architecture:**
        - Input, hidden, and output layers.
        - Weight initialization.
    - **Forward and Backward Propagation:**
        - How neural networks learn.
        - Chain rule and gradient computation.
    - **Training Neural Networks:**
        - Hands-on implementation in Python using frameworks like TensorFlow or PyTorch.

Class 3: Deep Learning Frameworks and Tools

- **Learning Objectives:**
    - Get introduced to popular deep learning frameworks.
    - Set up a development environment for deep learning.
    - Learn basic operations in TensorFlow, PyTorch, and Keras.
    - Build and train a simple model using Python.
- **Topics:**
    - **Introduction to Frameworks:**
        - TensorFlow, PyTorch, and Keras: Features and use cases.
    - **Environment Setup:**
        - Installing libraries and dependencies.
        - GPU vs. CPU for deep learning.
    - **Basic Operations:**
        - Tensor operations, matrix multiplications, and data handling.
    - **Model Creation:**
        - Building a simple neural network model in Python.

Class 4: Training Deep Learning Models

- **Learning Objectives:**
    - Learn how to preprocess and prepare data for training.
    - Understand loss functions and optimization algorithms.
    - Explore gradient descent variants and their applications.
    - Learn about overfitting, underfitting, and regularization techniques.
- **Topics:**
    - **Data Preparation:**
        - Data import, cleaning, and preprocessing.
        - Splitting data into training, validation, and test sets.
    - **Loss Functions:**
        - Mean Squared Error (MSE), Cross-Entropy Loss, etc.

- **Optimization Algorithms:**
  - Gradient Descent, Momentum, Nesterov Momentum, AdaGrad, RMSProp, Adam, Nadam.
- **Gradient Problems:**
  - Vanishing and exploding gradients.
- **Overfitting and Underfitting:**
  - Causes and solutions.
- **Regularization Techniques:**
  - L1/L2 regularization, dropout, early stopping.

Module 2: Computer Vision (8 Classes)

Class 5: Introduction to Computer Vision

- **Learning Objectives:**
  - Understand the tasks and applications of computer vision.
  - Learn how to handle image data.
  - Explore data augmentation techniques.
- **Topics:**
  - **Computer Vision Tasks:**
    - Image classification, object detection, segmentation, etc.
  - **Image Data Handling:**
    - Loading, resizing, and normalizing images.
  - **Data Augmentation:**
    - Techniques like rotation, flipping, cropping, and color jittering.

Class 6: Convolutional Neural Networks (CNNs)

- **Learning Objectives:**
  - Understand the architecture and components of CNNs.
  - Learn about convolution and pooling layers.
  - Implement a CNN in Python.
- **Topics:**
  - **CNN Architecture:**
    - Convolutional layers, pooling layers, and fully connected layers.
  - **Convolution and Pooling:**
    - Filters, strides, padding, and feature maps.
  - **Implementation:**
    - Building a CNN model using TensorFlow or PyTorch.

Class 7: Advanced CNN Architectures

- **Learning Objectives:**
  - Explore popular CNN models like LeNet, AlexNet, VGG, ResNet, and Inception.

    - Learn about transfer learning and fine-tuning.
- **Topics:**
    - **Popular CNN Models:**
        - LeNet, AlexNet, VGG, ResNet, Inception.
    - **Transfer Learning:**
        - Using pre-trained models for new tasks.
    - **Fine-Tuning:**
        - Adapting pre-trained models to specific datasets.

Class 8: Object Detection and Localization

- **Learning Objectives:**
    - Learn about object detection techniques like R-CNN, Fast R-CNN, Faster R-CNN, and YOLO.
    - Implement object detection models.
- **Topics:**
    - **Object Detection Techniques:**
        - R-CNN, Fast R-CNN, Faster R-CNN, YOLO.
    - **Implementation:**
        - Hands-on implementation of YOLO for object detection.

Class 9: Semantic Segmentation and Image Segmentation

- **Learning Objectives:**
    - Understand semantic segmentation techniques like U-Net and Fully Convolutional Networks.
    - Implement segmentation models.
- **Topics:**
    - **Segmentation Techniques:**
        - U-Net, Fully Convolutional Networks.
    - **Implementation:**
        - Practical examples and use cases.

Class 10: Generative Adversarial Networks (GANs) in Computer Vision

- **Learning Objectives:**
    - Understand the architecture and training of GANs.
    - Implement a GAN for image generation.
- **Topics:**
    - **GAN Architecture:**
        - Generator and discriminator networks.
    - **Training GANs:**
        - Challenges and best practices.
    - **Implementation:**
        - Building a GAN in Python.

Class 11: Applications of GANs in Computer Vision

- **Learning Objectives:**
  - Explore applications of GANs like image generation, style transfer, and super-resolution.
  - Implement GAN-based projects.
- **Topics:**
  - **Applications:**
    - Image generation, style transfer, super-resolution.
  - **Implementation:**
    - Hands-on projects using GANs.

Class 12: Computer Vision Projects

- **Learning Objectives:**
  - Implement a real-world computer vision project.
  - Learn best practices and troubleshooting techniques.
- **Topics:**
  - **Project:**
    - Automatic Dhaka Traffic Detection using the YOLO Model.
  - **Best Practices:**
    - Model evaluation, debugging, and optimization.

Module 3: Natural Language Processing (NLP) (7 Classes)

Class 13: Introduction to NLP

- **Learning Objectives:**
  - Understand NLP tasks and text preprocessing techniques.
  - Learn about regex and its applications in NLP.
- **Topics:**
  - **NLP Tasks:**
    - Text classification, sentiment analysis, etc.
  - **Text Preprocessing:**
    - Tokenization, stemming, lemmatization, stopword removal.
  - **Regex:**
    - Pattern matching and text extraction.

Class 14: Word Embeddings and Representations

- **Learning Objectives:**
  - Learn about word embeddings like TF-IDF, Word2Vec, GloVe, and FastText.
  - Explore contextual embeddings like ELMo and BERT.

- **Topics:**
  - **Word Embeddings:**
    - TF-IDF, Word2Vec, GloVe, FastText.
  - **Contextual Embeddings:**
    - ELMo, BERT.
  - **Implementation:**
    - Hands-on implementation in Python.

Class 15: Recurrent Neural Networks (RNNs) and Variants

- **Learning Objectives:**
  - Understand the architecture of RNNs, LSTMs, and GRUs.
  - Implement RNN-based models for NLP tasks.
- **Topics:**
  - **RNN Architecture:**
    - Basic RNN, LSTM, GRU.
  - **Implementation:**
    - Building RNN models in Python.

Class 16: Attention Mechanisms and Transformers

- **Learning Objectives:**
  - Understand the attention mechanism and transformer architecture.
  - Learn about input embeddings, positional encodings, and encoder-decoder layers.
- **Topics:**
  - **Attention Mechanism:**
    - Self-attention and multi-head attention.
  - **Transformers:**
    - Input embeddings, positional encodings, encoder, decoder, output layer.

Class 17: Advanced Transformer Models

- **Learning Objectives:**
  - Explore advanced transformer models like BERT, GPT, and T5.
  - Learn how to fine-tune pre-trained transformers.
- **Topics:**
  - **Transformer Models:**
    - BERT, GPT, T5.
  - **Fine-Tuning:**
    - Adapting pre-trained models for specific tasks.

Class 18: Sequence-to-Sequence Models and Applications

- **Learning Objectives:**
  - Understand sequence-to-sequence models for tasks like machine translation and text summarization.
  - Implement seq2seq models in Python.
- **Topics:**
  - **Applications:**
    - Machine translation, text summarization.
  - **Implementation:**
    - Hands-on projects using seq2seq models.

Class 19: NLP Projects

- **Learning Objectives:**
  - Implement a real-world NLP project.
  - Learn best practices and troubleshooting techniques.
- **Topics:**
  - **Project:**
    - Word Spelling Correction.
  - **Best Practices:**
    - Model evaluation, debugging, and optimization.

Module 4: Generative AI (6 Classes)

Class 20: Introduction to Generative AI

- **Learning Objectives:**
  - Understand the concept of generative models and their applications.
  - Explore use cases of generative AI in various industries.
- **Topics:**
  - **Generative Models:**
    - Overview and types.
  - **Applications:**
    - Image generation, text generation, music generation.

Class 21: Variational Autoencoders (VAEs)

- **Learning Objectives:**
  - Understand the architecture and training of VAEs.
  - Implement VAEs for image and text generation.
- **Topics:**
  - **VAE Architecture:**
    - Encoder, decoder, and latent space.
  - **Applications:**
    - Image and text generation.

    - **Implementation:**
        - Hands-on implementation in Python.

Class 22: Advanced GAN Techniques

- **Learning Objectives:**
    - Explore advanced GAN variants like DCGAN, CycleGAN, and StyleGAN.
    - Learn about training stability and challenges.
- **Topics:**
    - **GAN Variants:**
        - DCGAN, CycleGAN, StyleGAN.
    - **Training Challenges:**
        - Mode collapse, instability.
    - **Implementation:**
        - Hands-on projects using advanced GANs.

Class 23: Generative AI in NLP

- **Learning Objectives:**
    - Learn about LangChain and Hugging Face for building LLM-powered applications.
    - Fine-tune LLM models like Gemma and LLAMA.
    - Implement text generation and chatbot applications.
- **Topics:**
    - **LangChain & Hugging Face:**
        - Introduction and use cases.
    - **LLM Models:**
        - Fine-tuning Gemma and LLAMA models.
    - **Applications:**
        - Chatbots, content creation.
    - **Implementation:**
        - Hands-on projects using LLMs.

Class 24: Generative AI Project

- **Learning Objectives:**
    - Implement a real-world generative AI project.
    - Learn best practices and troubleshooting techniques.
- **Topics:**
    - **Project:**
        - LLM LangChain Project using Vector Database.
    - **Best Practices:**
        - Model evaluation, debugging, and optimization.

This detailed course outline provides a comprehensive roadmap for learning deep learning and generative AI, with a mix of theoretical concepts and

hands-on projects. Each class builds on the previous one, ensuring a gradual and thorough understanding of the topics.