Detailed syllabus **“Become Job Ready Data Scientist”** course:

### **1. Python**

### **Week 1: Python Basics and Foundational Concepts**

* **Introduction to Python**
  + **Syntax, indentation, comments, understanding Python's interpreter**
* **Variable Naming and Datatypes**
  + **Naming conventions, numeric types (int, float, complex), strings, type conversion, type hints**
* **Basic Input/Output**
  + **Using input() and print(), string formatting (f-strings, .format())**

### **Week 2: Control Flow and Collections**

* **Control Flow**
  + **Conditional statements (if-else, nested conditions), logical operators (and, or, not)**
  + **Loops: for and while loops, break, continue, else with loops**
* **Data Structures: Lists and Tuples**
  + **Creating lists, list indexing, slicing, list methods (append, pop, extend, sort)**
  + **Tuples: Immutable properties, tuple unpacking, practical use cases**

### **Week 3: Advanced Data Structures and Comprehensions**

* **Dictionaries**
  + **Creating dictionaries, accessing/updating/deleting keys, dictionary methods (keys(), values(), items())**
  + **Nested dictionaries, practical examples (JSON-like data)**
* **Sets**
  + **Creating sets, set operations (union, intersection, difference, symmetric\_difference)**
* **List/Dictionary Comprehensions**
  + **Using comprehensions to create lists and dictionaries efficiently, conditional comprehensions**

### **Week 4: Functions and Modules**

* **Functions**
  + **Defining functions, parameters and arguments, default arguments, return statements**
  + **Anonymous functions (lambda), function annotations, scope (local, global, nonlocal)**
* **Modules and Packages**
  + **Importing modules (import, from ... import), using standard libraries (math, random, os)**
  + **Working with custom modules and \_\_init\_\_.py**

### **Week 5: File Handling and Error Management**

* **File Handling**
  + **Reading and writing text files (open(), read(), write(), file modes r, w, a)**
  + **Working with CSV and JSON files (csv module, json module)**
* **Error and Exception Handling**
  + **Understanding exceptions, using try, except, finally, raising custom exceptions (raise)**

### **Week 6: Python for Data Analysis**

* **Introduction to NumPy**
  + **Arrays vs. lists, creating arrays, array operations, broadcasting, indexing/slicing, reshaping arrays**
* **Introduction to pandas**
  + **Series and DataFrames, creating DataFrames, basic operations (head(), describe(), info()), indexing and slicing and other methods**
* **Basic Data Visualization**
  + **Introduction to matplotlib and seaborn, creating line plots, bar charts, and scatter plots**

### **Week 7-8: Advanced Python Concepts and Project**

* **Object-Oriented Programming (OOP)**
  + **Classes and objects, attributes and methods, inheritance, polymorphism**
* **Python Project**
  + **Exploratory Data Analysis (EDA) project using pandas, NumPy, matplotlib, and seaborn**

### **2. Machine Learning**

### **Week 1: Introduction to Machine Learning and Data Preprocessing**

* **Introduction to Machine Learning**
  + **Definition and importance, types of ML (supervised, unsupervised, reinforcement learning)**
  + **Real-world applications: recommendation systems, fraud detection, sentiment analysis**
* **ML Workflow**
  + **Problem definition, data collection, preprocessing, modeling, evaluation, and deployment**
* **Data Preprocessing**
  + **Handling missing data (mean/mode/median imputation, forward/backward fill)**
  + **Encoding categorical variables (one-hot encoding, label encoding)**
  + **Feature scaling (min-max scaling, standardization, normalization)**
* **Exploratory Data Analysis (EDA)**
  + **Identifying outliers, correlation analysis, visualizing distributions**

### **Week 2: Supervised Learning - Regression**

* **Introduction to Regression**
  + **Regression vs. classification, when to use regression models**
* **Linear Regression**
  + **Mathematical intuition, assumptions, cost function (MSE), gradient descent**
  + **Implementation with scikit-learn (LinearRegression), interpreting coefficients**
* **Polynomial Regression**
  + **When linearity doesn’t fit, polynomial feature transformation, overfitting and underfitting**
* **Regularization Techniques**
  + **Ridge regression (L2 regularization), Lasso regression (L1 regularization)**
* **Model Evaluation Metrics**
  + **Mean Absolute Error (MAE), Mean Squared Error (MSE), R² score**

### **Week 3: Supervised Learning - Classification**

* **Introduction to Classification**
  + **Binary vs. multi-class classification, decision boundaries**
* **Logistic Regression**
  + **Mathematical intuition, sigmoid function, decision threshold tuning**
  + **Practical implementation using scikit-learn**
* **K-Nearest Neighbors (KNN)**
  + **Distance metrics (Euclidean, Manhattan), choosing k, bias-variance tradeoff**
* **Decision Trees**
  + **Splitting criteria (Gini impurity, entropy), visualizing decision trees**
* **Random Forests**
  + **Bagging, Boosting, feature importance, handling overfitting with ensemble methods**
* **Model Evaluation Metrics for Classification**
  + **Confusion matrix, precision, recall, F1-score, ROC-AUC**

### **Week 4: Unsupervised Learning and Advanced Topics**

* **Clustering Techniques**
  + **K-Means: Centroid initialization, elbow method for determining k**
  + **Hierarchical Clustering: Agglomerative clustering, dendrograms**
  + **DBSCAN: Density-based clustering, epsilon and min\_samples**
* **Dimensionality Reduction**
  + **PCA (Principal Component Analysis): Explained variance, feature transformation**
  + **t-SNE (t-Distributed Stochastic Neighbor Embedding): Visualization of high-dimensional data**
* **Anomaly Detection**
  + **Techniques for outlier detection, applications in fraud detection and system monitoring**

### **Week 5: Model Optimization and Deployment**

* **Model Optimization**
  + **Cross-validation: K-fold, leave-one-out, stratified K-fold**
  + **Hyperparameter Tuning: Grid search, random search, Bayesian optimization**
  + **Handling imbalanced datasets: SMOTE, weighted loss functions**
  + **Model compression**
* **Introduction to ML Deployment**
  + **Saving and loading models (joblib, pickle)**
  + **Basic Flask app for serving ML models**
* **End-to-End ML Pipeline**
  + **Combining preprocessing, model building, evaluation, and deployment using scikit-learn pipelines**

**Project:**

1. **Regression Project: Predicting house prices using linear and polynomial regression.**
2. **Classification Project: Customer churn prediction using logistic regression, decision trees, and random forests.**

### **3. Deep Learning**

### **Week 1: Introduction to Deep Learning and Neural Networks**

* **Introduction to Deep Learning**
  + **Why deep learning? Real-world applications (image recognition, NLP, autonomous vehicles)**
  + **Difference between machine learning and deep learning**
  + **Overview of deep learning frameworks (TensorFlow, PyTorch)**
* **Neural Networks Basics**
  + **Biological vs. artificial neurons**
  + **Architecture: Input, hidden, and output layers**
  + **Activation functions: ReLU, sigmoid, softmax, tanh**
  + **Cost functions: MSE, cross-entropy**
* **Feedforward Neural Networks**
  + **Forward and backward propagation, gradient descent, learning rate tuning**
  + **Hands-on: Building a simple neural network from scratch in TensorFlow/Keras**

### **Week 2: Convolutional Neural Networks (Image-Based Project)**

* **Introduction to Convolutional Neural Networks (CNNs)**
  + **Convolution operation, filters/kernels, feature maps, auto-encoding**
  + **Pooling layers (max pooling, average pooling), flattening, fully connected layers**
* **Architectures and Use Cases**
  + **Famous architectures: VGG, ResNet, EfficientNet (brief overview)**
  + **Common use cases: Object detection, image classification**
* **Project (Part 1): Image Classification**
  + **Dataset: CIFAR-10 or Fashion-MNIST**
  + **Preprocessing: Resizing, normalizing, data augmentation (rotation, flipping)**
  + **Building a CNN model using TensorFlow/Keras**
  + **Training, validation, and testing the model**

### **Week 3: Optimization and Transfer Learning (Image-Based Project Continuation)**

* **Optimization Techniques**
  + **Optimizers: SGD, Adam, RMSProp**
  + **Regularization: Dropout, L2 regularization**
  + **Learning rate schedules: Step decay, exponential decay**
* **Transfer Learning**
  + **Concept and benefits of transfer learning**
  + **Using pre-trained models (e.g., ResNet, MobileNet)**
  + **Fine-tuning and feature extraction**
* **Project (Part 2): Improving Image Classification Model**
  + **Apply transfer learning to improve accuracy**
  + **Fine-tune pre-trained model (e.g., MobileNet)**
  + **Evaluate and interpret the model's performance**

### **Week 4: Recurrent Neural Networks (RNNs) and Natural Language Processing (NLP-Based Project)**

* **Introduction to RNNs**
  + **Sequential data, limitations of feedforward networks**
  + **RNN structure, vanishing gradient problem**
* **Advanced RNNs: LSTMs and GRUs**
  + **Long Short-Term Memory (LSTM): Gates (forget, input, output), cell state**
  + **Gated Recurrent Units (GRUs): Simplified LSTM architecture**
* **Basics of Natural Language Processing**
  + **Tokenization, stemming, lemmatization, word embeddings (Word2Vec, GloVe)**
  + **Text preprocessing: Removing stopwords, handling punctuation, lowercasing**
* **Project (Part 1): Sentiment Analysis on Text Data**
  + **Dataset: IMDb or Twitter Sentiment Dataset**
  + **Preprocessing: Tokenization, padding sequences**
  + **Building an RNN/LSTM model for sentiment classification**

### **Week 5: Transformers and Advanced NLP Techniques (NLP-Based Project Continuation)**

* **Introduction to Transformers**
  + **Concept of attention, encoder-decoder architecture, multi-head attention**
  + **Overview of BERT, GPT, and their applications in NLP**
* **Fine-Tuning Pre-Trained NLP Models**
  + **Hugging Face library: Loading and fine-tuning BERT or GPT for text classification**
  + **Handling large datasets and transfer learning for NLP**
* **Project (Part 2): Improving NLP Sentiment Analysis**
  + **Fine-tune BERT/GPT on the same sentiment dataset**
  + **Compare performance with RNN/LSTM model**
  + **Model evaluation: Precision, recall, F1-score, confusion matrix**

### **Key Deliverables**

1. **Project 1 (Image-Based): Image classification model with CNNs, enhanced using transfer learning.**
2. **Project 2 (NLP-Based): Sentiment analysis model with LSTMs, improved using a fine-tuned transformer model like BERT.**

### **4. Generative AI**

### **Week 1: Foundations of Generative AI and GANs**

* **Introduction to Generative AI**
  + What is Generative AI? Applications: Image generation, text generation, deepfakes, AI art
  + Difference between discriminative and generative models
  + Generative AI in industries: Gaming, healthcare, content creation
* **Introduction to Generative Adversarial Networks (GANs)**
  + Architecture: Generator and discriminator
  + Adversarial training, loss functions, mode collapse
* **Hands-on Project (Part 1): Building a Basic GAN**
  + Dataset: MNIST (handwritten digits)
  + Preprocessing: Data normalization
  + Implementing a simple GAN using PyTorch or TensorFlow
  + Training the GAN and generating basic images

### **Week 2: Advanced GANs and Variational Autoencoders (VAEs)**

* **Advanced GAN Concepts**
  + Conditional GANs (cGANs): Generating specific categories (e.g., numbers, colors)
  + CycleGANs: Style transfer for unpaired image datasets (e.g., turning photos into paintings)
  + Progressive GANs and StyleGANs: High-resolution image generation
* **Introduction to Variational Autoencoders (VAEs)**
  + Difference between autoencoders and VAEs
  + Understanding the encoder-decoder structure and latent space
  + Applications of VAEs: Image reconstruction, anomaly detection
* **Hands-on Project (Part 2): Image-to-Image Translation**
  + Dataset: Edges-to-shoes or similar paired datasets
  + Implementing a CycleGAN for style transfer or image-to-image mapping

### **Week 3: Generative AI for Text and Language Models**

* **Generative Pre-trained Transformers (GPT)**
  + Architecture: Transformers, self-attention mechanism, positional encoding
  + Overview of GPT models (GPT-2, GPT-3, GPT-4) and their capabilities
* **Fine-Tuning Pre-trained Language Models**
  + Using Hugging Face library for fine-tuning GPT models
  + Practical applications: Chatbots, text summarization, content generation
* **Hands-on Project (Part 3): Text Generation**
  + Dataset: News articles or conversational datasets
  + Fine-tuning GPT on a custom dataset for text completion or dialogue generation
  + Evaluating text quality: Perplexity, human evaluation

### **Week 4: Advanced Generative AI Techniques and Applications**

* **Diffusion Models**
  + Understanding diffusion models (e.g., Stable Diffusion, DALL-E)
  + Applications in text-to-image generation and AI art creation
* **Ethics and Challenges in Generative AI**
  + Bias in generated outputs, intellectual property issues, misuse of generative models (e.g., deepfakes)
  + Best practices and tools for ethical AI development
* **Hands-on Project (Part 4): Text-to-Image Generation**
  + Tools: Stable Diffusion, OpenAI’s DALL-E API
  + Generating images based on textual descriptions
  + Fine-tuning image generation models on custom datasets for specific styles

### **Key Deliverables**

1. **Project 1**: Basic GAN for image generation (e.g., MNIST handwritten digits).
2. **Project 2**: CycleGAN for style transfer or image-to-image translation.
3. **Project 3**: Fine-tuned GPT for text generation (e.g., creating human-like dialogues or articles).
4. **Project 4**: Text-to-image generation using Stable Diffusion or DALL-E.

### **5. Building AI Agents**

### **Week 1: Introduction to AI Agents and LangChain Basics**

* **What Are AI Agents?**
  + **Definition and purpose of AI agents**
  + **Applications in customer service, task automation, and personalized recommendations**
  + **Key components: Perception, decision-making, and action**
* **Introduction to LangChain**
  + **What is LangChain? Importance of modular frameworks for building LLM-powered applications**
  + **Core concepts:**
    - **Chains: Sequential and parallel workflows**
    - **Agents: Decision-making systems powered by LLMs**
    - **Tools: Integrating APIs, web scraping, and database queries**
* **Hands-on Project (Part 1): Task Automation with LangChain**
  + **Use case: Automating data retrieval and summarization from multiple online sources**
  + **Implementing:**
    - **Chain for querying APIs and processing results**
    - **LLM-based summarization with OpenAI models**
    - **Tool integration (e.g., Google Search API)**

### **Week 2: Advanced AI Agent Concepts and CrewAI Project**

* **Advanced AI Agent Design**
  + **Memory in AI agents: Types (short-term, long-term), vector stores, and persistence with LangChain**
  + **Decision-making: Action planning using tools like Python functions and structured APIs**
  + **Error handling: Managing unexpected outputs or failures gracefully**
* **Introduction to CrewAI**
  + **Overview of CrewAI: A simplified tool for building AI agents without deep coding knowledge**
  + **How CrewAI complements LangChain for rapid prototyping and iterative development**
* **Hands-on Project (Part 2): Task-Oriented AI Agent with CrewAI and LangChain**
  + **Use case: Building an AI assistant for project management tasks**
  + **Features to implement:**
    - **Querying deadlines and project updates from a spreadsheet**
    - **Summarizing team performance using pre-defined metrics**
    - **Suggesting next steps or optimizations for workflows**
  + **Tools and integrations:**
    - **Google Sheets API for task data**
    - **OpenAI or Hugging Face LLMs for summarization and recommendations**
    - **Memory modules to maintain conversation context**

### **Key Deliverables**

1. **Project 1 (LangChain): AI task automation agent for real-world data retrieval and summarization.**
2. **Project 2 (CrewAI + LangChain): Task-oriented AI agent for project management, combining multiple tools and memory for enhanced decision-making.**

### **6. SQL**

### **Week 1: Introduction to SQL and Database Design**

* **Understanding Databases**
  + Relational databases vs. non-relational databases
  + Database management systems (DBMS): MySQL, PostgreSQL, SQLite
  + Introduction to tables, rows, columns, and relationships
* **Basic SQL Queries**
  + SELECT statement, filtering with WHERE clause
  + Sorting results with ORDER BY
  + Aggregating data: COUNT, SUM, AVG, MAX, MIN
* **Basic Joins**
  + Inner join, left join, right join, full outer join
  + Join conditions and understanding relationships
* **Hands-on Project (Part 1): Data Extraction from a Sample Dataset**
  + Dataset: Sales data, customer data, or ecommerce data
  + Extracting relevant data using SELECT and filtering with WHERE
  + Sorting results, performing basic aggregation (e.g., total sales, customer counts)

### **Week 2: Advanced SQL Queries and Subqueries**

* **Advanced Joins**
  + Self-joins, cross joins
  + Combining multiple tables with complex join conditions
* **Subqueries**
  + Subqueries in SELECT, WHERE, and FROM clauses
  + Using subqueries for filtering and calculating derived metrics
* **Grouping Data**
  + GROUP BY, HAVING clause for advanced aggregation
  + Using aggregate functions in grouped data (e.g., total sales by region, average order size by product)
* **Hands-on Project (Part 2): Building Reports with Advanced Queries**
  + Dataset: Sales and customer transactions
  + Building reports with GROUP BY to analyze total sales by region, average transaction value
  + Using subqueries to identify top-performing products, customers, or regions

### **Week 3: Data Modification and Indexing**

* **Data Insertion, Updates, and Deletions**
  + INSERT INTO statement for adding records
  + UPDATE statement for modifying data, handling data integrity
  + DELETE statement and handling referential integrity (e.g., cascading deletes)
* **Transactions and ACID Properties**
  + Understanding transactions (COMMIT, ROLLBACK)
  + Ensuring data consistency and atomicity
* **Indexes and Query Optimization**
  + Importance of indexing for fast query performance
  + Creating and using indexes
  + Query execution plans and optimizing complex queries
* **Hands-on Project (Part 3): Data Management and Optimization**
  + Dataset: Employee records and project assignments
  + Inserting, updating, and deleting records while maintaining integrity
  + Optimizing queries using indexes, improving performance on large datasets
  + Analyzing query performance using execution plans

### **Week 4: Complex SQL Queries, Views, and Company-Specific Problems**

* **Complex Queries**
  + Window functions (e.g., ROW\_NUMBER(), RANK(), LEAD(), LAG())
  + Working with dates and time intervals (DATE, TIMESTAMP, DATE\_DIFF, DATE\_ADD)
  + String manipulation and pattern matching (LIKE, REGEXP)
* **Views and Stored Procedures**
  + Creating and using views for reusable queries
  + Introduction to stored procedures for automating SQL logic
  + Triggers and events for automated actions based on data changes
* **Company-Specific Problem Solving**
  + Solving real-world company-specific SQL problems based on the industry (e.g., e-commerce, finance, healthcare)
    - Example: Analyzing customer churn in an e-commerce company
    - Example: Financial report generation for a company
    - Example: Analyzing employee performance and project completion in a corporate environment
* **Hands-on Project (Part 4): Company-Specific SQL Project**
  + Use case: Building complex reports and automating data retrieval for business operations
  + Working with company datasets to solve industry-specific problems using SQL queries, joins, and optimizations

### **Key Deliverables**

1. **Project 1**: Basic SQL report building with dataset extraction, filtering, and aggregation.
2. **Project 2**: Advanced report generation with multiple joins, subqueries, and groupings.
3. **Project 3**: Data management, transaction handling, and query optimization using indexing and execution plans.
4. **Project 4**: Solving a company-specific SQL problem (e.g., customer churn analysis, financial report generation).

### **7. Capstone Projects (+2 Projects)**

**End-to-End ML Pipeline**

* + Data preprocessing, feature engineering, model building, evaluation, deployment