**Technical Experience Prerequisite**

Students are expected to have:

* **Programming Knowledge:**
  + Familiarity with any other programming language like **C#, JavaScript, or .NET**
* **Web Development Basics:**
  + Understanding of **HTML**, **CSS**, and **JavaScript** fundamentals.
  + Prior experience working on **front-end frameworks** (like Angular or React).
* **Database Knowledge:**
  + Basic understanding of **SQL commands** and relational database concepts like tables, relationships, and CRUD operations.
* **Development Environment Familiarity:**
  + Experience working with **IDEs** like Visual Studio Code, PyCharm, or any equivalent.
  + Basic command-line knowledge for navigating and executing scripts.

**Expectations from Students**

To make the most out of this training, students are expected to:

* **Participate Actively:**
  + Engage in discussions and ask questions for clarity.
  + Share relevant experiences or challenges faced in past projects.
* **Complete Hands-On Activities:**
  + Follow along with instructor-led coding sessions.
  + Attempt assigned tasks during lab sessions.
* **Understand Lab Environment in Advance:**
  + Connect and validate the labs before training begins.
* **Self-Review:**
  + Go through the provided study material after each session.

**What is Not Covered in Training**

1. **Advanced Deep Learning Concepts:**
   * Deep learning frameworks like TensorFlow or PyTorch.
   * Detailed discussion of algorithms like neural networks.
2. **Front-End Framework Development:**
   * No focus on Angular, React, or Vue.js.
   * Only basic HTML, CSS, and JavaScript for templates.
3. **DevOps Integration:**
   * No CI/CD pipeline setup or deployment using cloud services.
4. **Cloud-Based Deployments:**
   * No in-depth deployment to platforms like AWS, Azure, or GCP.
   * Deployment covered will be limited to local or basic hosting.
5. **Performance Optimization:**
   * No advanced optimization techniques for web apps or ML models.

**Course Content**

**Session 1:- 10:30am to 12:30pm**

**Session 2:- 01:30pm to 03:30pm**

**Session 3:- 04:00pm to 06:00pm**

**Day 1: SQL Database Connectivity with Python & Django Basics**

1. **Session 1 - Introduction to SQL and Relational Databases (1 hour)**
   * SQL basics: SELECT, INSERT, UPDATE, DELETE
   * Overview of RDBMS and MS SQL
   * **Q&A (15 minutes)**
2. **Session 1 and 2 - Python Database Connectivity with MS SQL (2.5 hours)**
   * Setting up Python and MS SQL connectivity
   * Performing CRUD operations using Python
   * Hands-on: Execute SQL queries from Python
   * **Q&A (15 minutes)**
3. **Session 2 & 3 - Introduction to Django Framework (2 hours)**
   * Overview of Django and its features
   * Setting up a Django project
   * Hands-on: Create a basic Django project
   * **Q&A (15 minutes)**
4. **Deployment overview of Python projects (30 minutes)**

**Day 2: Building Web Applications with Django**

1. **Session 1 - Django Models (1.5 hours)**
   * Introduction to Django ORM
   * Creating models and migrations
   * Hands-on: Build and interact with models
   * **Q&A (15 minutes)**
2. **Session 1 - Forms and User Input (1.5 hours)**
   * Handling forms and user input in Django
   * Validating and processing data
   * Hands-on: Create a form-based Django app
   * **Q&A (15 minutes)**
3. **Session 2 - CRUD Operations with Django (2 hours)**
   * Saving and retrieving form data in the database
   * Implementing CRUD functionality in Django projects
   * Hands-on: Implement CRUD functionality
   * **Q&A (15 minutes)**
4. **Session 3 - Mini Project and Wrap-Up (1 hour)**
   * Consolidate all concepts into a mini Django project
   * **Q&A (15 minutes)**

**Day 3: Advanced Django Features and Flask Basics**

1. **Session 1 - User Authentication in Django (1.5 hours)**
   * Implementing user registration, login, and logout functionality
   * Hands-on: Build a user authentication system
   * **Q&A (15 minutes)**
2. **Session 1 - Introduction to Flask (1.5 hours)**
   * Setting up Flask and building a basic web application
   * URL routing and handling requests
   * Hands-on: Create a simple Flask app
   * **Q&A (15 minutes)**
3. **Session 2 - Jinja2 Templating (1 hour)**
   * Using Jinja2 for dynamic HTML rendering in Flask
   * Hands-on: Create a Flask app with templates
   * **Q&A (15 minutes)**
4. **Session 3 - Mini Project: Flask Web App (1 hour)**
   * Build a basic CRUD app in Flask
   * **Q&A (15 minutes)**
5. **Deploy project to localhost and access it from the network (1 Hour)**

**Day 4: APIs and Machine Learning Basics**

1. **Session 1 - Creating APIs with Flask (1.5 hours)**
   * Introduction to REST APIs
   * Building APIs using Flask
   * Hands-on: Create an API to handle basic operations
   * **Q&A (15 minutes)**
2. **Session 2 - Publishing APIs for Machine Learning Models (1.5 hours)**
   * Integrating ML algorithms into Flask APIs
   * Hands-on: Publish a Flask API for ML
   * **Q&A (15 minutes)**
3. **Session 2 and 3 - Calling APIs (1.5 hours)**
   * Using Postman and Python to call APIs
   * Hands-on: Test APIs with tools and scripts
   * **Q&A (15 minutes)**
4. **Session 3 - Overview of Machine Learning Concepts (1.5 hours)**
   * Introduction to supervised and unsupervised learning
   * Common ML algorithms: Linear regression, classification, clustering
   * Hands-on: Understand ML workflows
   * **Q&A (15 minutes)**

**Day 5: Pandas and Data Visualization with Matplotlib / Seaborn**

1. **Session 1 - Introduction to Pandas (30 Minutes)**
   * Understanding the importance of Pandas in data analysis
   * Creating and manipulating DataFrames
   * Data exploration: Overview of Series and DataFrames
   * Hands-on: Explore datasets using Pandas
   * **Q&A (5 minutes)**
2. **Session 1 - Reading and Writing Data(30 Minutes)**
   * Reading and writing data in CSV, Excel, and SQL formats
   * Practical use cases of importing/exporting data
   * Hands-on: Load data from files and save processed data
   * **Q&A (10 minutes)**
3. **Session 1 - Data Cleaning and Preprocessing(30 Minutes)**
   * Handling missing values and duplicates
   * Filtering, sorting, and grouping data
   * Real-world Data Manipulation case study
   * Hands-on: Perform cleaning and preprocessing tasks with Pandas
   * **Q&A (10 minutes)**
4. **Session 1 - Introduction to Data Visualization (45 minutes)**
   * Importance and use cases of data visualization
   * Real-world applications
   * **Q&A (15 minutes)**
5. **Session 2 - Introduction to Matplotlib (1.5 hours)**
   * Overview of Matplotlib
   * Basic plots: Line plots, bar charts, histograms
   * Hands-on: Create basic plots
   * **Q&A (15 minutes)**
6. **Session 3 - Customizing Plots (1 hours)**
   * Adding titles, labels, legends, and annotations
   * Customizing colours, line styles, and markers
   * Hands-on: Customize visualizations
   * **Q&A (15 minutes)**
7. **Session 3 - Subplots and Advanced Plotting using Seaborn (1 hour)**
   * Introduction to Seaborn
   * Creating subplots and multiple plots on the same graph
   * Hands-on: Design advanced multi-plot layouts
   * **Q&A (15 minutes)**

**Day 6: Machine Learning Integration and Advanced Concepts**

1. **Session 1 - Building Simple ML Models (2 hours)**
   * Data preprocessing with Pandas and NumPy
   * Training and testing ML models using Scikit-learn
   * Model evaluation metrics (accuracy, precision, recall)
   * Hands-on: Train and evaluate a simple ML model
   * **Q&A (15 minutes)**
2. **Session 2 - Integrating ML Models into Web Applications (2 hours)**
   * Using Pickle/Joblib to save models
   * Embedding ML models into Flask/Django apps
   * Hands-on: Deploy ML functionality in a web app
   * **Q&A (15 minutes)**
3. **Session 3 - Data Visualization in Web Applications (1 hour)**
   * Using Matplotlib or Plotly to embed charts
   * Hands-on: Visualize ML results in a web app
   * **Q&A (15 minutes)**
4. **Session 3 - Exploring Advanced ML Algorithms (1 hour)**
   * Introduction to decision trees, random forests, and clustering models
   * Practical applications in a web context
   * **Q&A (15 minutes)**

**Day 7: Consolidation and Additional Topics**

1. **Session 1 - API Security and Best Practices (1.5 hours)**
   * Securing REST APIs
   * Adding authentication and authorization layers
   * Hands-on: Implement API security measures
   * **Q&A (15 minutes)**
2. **Session 1 - Advanced API Integration (1.5 hours)**
   * Testing APIs in different environments
   * Building and consuming APIs in a multi-app ecosystem
   * Hands-on: Create and test advanced API workflows
   * **Q&A (15 minutes)**
3. **Session 2 - End-to-End Project (2 hours)**
   * Consolidate Django, Flask, APIs, and ML concepts into a single project
   * Students will build a mini project integrating all topics
   * **Q&A (15 minutes)**
4. **Session 3 - Wrap-Up and Feedback (1 hour)**
   * Review key learnings and address open questions
   * Feedback from participants
   * Discuss next steps for applying the concepts