**Integrating PostgreSQL with Python for ML, and deploying it using tools like Streamlit and FastAPI.**

**1. Course Overview:**

**Objective:**

* Teach students how to build a car price prediction model using PostgreSQL as the database, Python for data processing, ML algorithms for predictions, and Streamlit + FastAPI for deployment.
* Provide hands-on experience working with a real-world dataset and deploying the solution, so students can showcase their skills with a complete project.

**2. Structuring the One-Month Course**

* You can break the four weeks into clear, practical, and achievable tasks:
* Week 1 – Introduction to PostgreSQL, Data Exploration, and Preprocessing

**PostgreSQL Basics:**

* Introduction to PostgreSQL (installation, creating a database, connecting with Python).
* Creating tables and managing data in PostgreSQL (working with your car\_sales\_data).
* Loading data from CSV into PostgreSQL using SQLAlchemy or psycopg2.

**Data Exploration & Preprocessing:**

* Connect to PostgreSQL from Python.
* Data cleaning using pandas: handling missing values, encoding categorical variables (Fuel\_Type, Seller\_Type, etc.), and scaling numerical features (Present\_Price, Kms\_Driven, etc.).

**Deliverables:**

* Create a table in PostgreSQL and import your dataset.
* Perform basic data cleaning and exploration.

**Week 2 – Model Development: Building ML Models**

**Understanding the Problem:**

* Discuss the problem of predicting Selling\_Price based on features like Car\_Name, Fuel\_Type, Kms\_Driven, etc.

**Machine Learning Model Building:**

* Linear Regression: Start with a simple regression model.
* Random Forest: Use a tree-based algorithm for better accuracy.
* Train and evaluate models using sklearn.
* Hyperparameter tuning.

Deliverables:

* Students will build, train, and evaluate two models (Linear Regression and Random Forest) using their data.
* Use SQL queries to extract the data for training and testing models.

**Week 3 – Deployment with FastAPI & Streamlit**

**FastAPI for Model Deployment:**

* Introduction to FastAPI: Create a RESTful API that exposes your ML model as a service.
* How to use FastAPI to make predictions with the trained model (e.g., accept car features as input, return predicted price).
* Streamlit for Building the Interface:
* Introduction to Streamlit: Build a simple, interactive UI where users can input car data and get price predictions.
* Use Streamlit to display model predictions in a simple web interface.

Deliverables:

* Deploy the model using FastAPI.
* Build a Streamlit app to interact with the model.

**Week 4 – Final Project & Review**

**Final Review:**

* Refine the project: Make improvements, handle errors, optimize performance.
* Deploy the model on Heroku or another cloud service.

**Capstone Project:**

* Students will deploy their own models, set up a personal portfolio, and showcase their project.

**Deliverables:**

* Fully deployed ML model with a Streamlit interface and a FastAPI backend.