□ Day 1: Data Collection, Preprocessing & EDA

Tools: Jupyter Notebook

□ ***Folder Structure (for now):***

```
loan_approval_project/
│
├── data/
│   └── train.csv  (Download from Kaggle)
├── notebook/
│   └── loan_approval_analysis.ipynb
```

□ Step 1: Data Collection (load dataset)

```
# loan_approval_analysis.ipynb

# Import required libraries
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv("../data/train.csv")

# Basic info
print("Shape:", df.shape)
df.head()
```

☑**Download dataset** from: https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset

---

□ Step 2: Data Preprocessing

```
# Check for missing values
df.isnull().sum()

# Fill missing categorical values with mode
for col in ['Gender', 'Married', 'Dependents', 'Self_Employed', 'Credit_History', 'Loan_Amount_Term']:
    df[col].fillna(df[col].mode()[0], inplace=True)

# Fill numerical missing value
df['LoanAmount'].fillna(df['LoanAmount'].median(), inplace=True)

# Confirm no missing values
df.isnull().sum()
```

□ Step 3: Encode Categorical Variables

```
# Encode categorical features using Label Encoding
from sklearn.preprocessing import LabelEncoder

cat_cols = ['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
        'Property_Area', 'Loan_Status']
le = LabelEncoder()
for col in cat_cols:
   df[col] = le.fit_transform(df[col])
```

□ Step 4: Exploratory Data Analysis (EDA)

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot target variable
sns.countplot(x='Loan_Status', data=df)
plt.title("Loan Status Distribution")
plt.show()

# Plot categorical features against Loan_Status
for col in ['Gender', 'Married', 'Education', 'Self_Employed', 'Property_Area']:
   sns.countplot(x=col, hue='Loan_Status', data=df)
   plt.title(f"{col} vs Loan Status")
   plt.show()

# Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='Blues')
plt.title("Correlation Matrix")
plt.show()
```

□ Step 5: Feature Scaling

```
. from sklearn.preprocessing import StandardScaler

# Separate features and target
X = df.drop(['Loan_ID', 'Loan_Status'], axis=1)
y = df['Loan_Status']

# Standardize numerical columns
num_cols = ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term']
scaler = StandardScaler()
X[num_cols] = scaler.fit_transform(X[num_cols])
```

☑Day 1 Target Checklist:

- ☑ Load and understand dataset
- ☑ Clean missing values
- ☑ Encode categorical variables

- ☑ Visualize key insights using EDA
- ☑ Prepare features for modeling