Project Design Phase-II
Technology Stack (Architecture & Stack)

Team ID	LTVIP2025TMID38181
Project Name	Enchanted Wings :Marvels Of butterfly Species
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example:

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

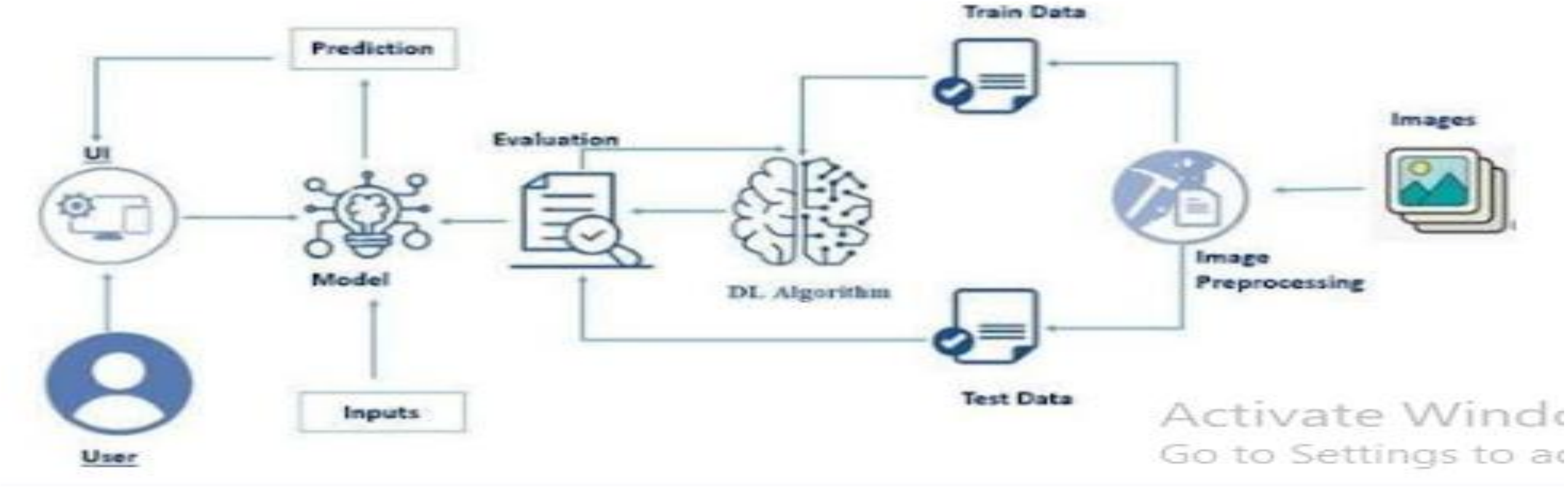

Table 1 : Components and Technologies Used in Butterfly Classification Project

S. No	Component / Project Step	Technology / Tool / Concept Used
1.	Data Collection	Kaggle dataset download / Kaggle API
2.	Data Pre-processing	Image resizing, normalization, cleaning
3.	Data Augmentation	Keras ImageDataGenerator (rotation, flipping, zooming)
4.	Data Splitting	Train-test split using sklearn or manually in folders
5.	Model Building	TensorFlow / Keras
6.	Transfer Learning	VGG16 Pre-trained Model
7.	Model Training & Testing	CNN training with optimizers like Adam or SGD
8.	Model Evaluation	Accuracy, Confusion Matrix, Loss Graphs
9.	Model Saving	Keras .h5 model save format
10.	Application Interface	Flask Framework
11.	Frontend Design	HTML/CSS for basic UI
12.	Visualization	Matplotlib, Seaborn, Random image viewer using IPython
13.	Understanding Concepts	DL, CNNs, Overfitting, Optimizers, Regularization

Table 2: Applications and Characteristics Used in Butterfly Classification Project

S. No	Application	Characteristics	Description
1.	Model Construction	Uses pre-trained VGG16	Base model with frozen weights used to extract image features.
2.	Regularization	Dropout layer added	Dropout reduces overfitting by randomly disabling neurons during training.
3.	Classification	Dense SoftMax layer	Final layer predicts one of the butterfly species based on image features.
4.	Compilation	Adam optimizer, Sparse Categorical Cross-Entropy	Efficient optimization and loss handling for multi-class classification.
5.	Training Process	15 epochs with callbacks	Includes ModelCheckpoint and EarlyStopping to save best model and prevent overfitting.
6.	Model Output	Saved as `.h5` file	Trained model saved as "`vgg16_model.h5`" for reuse or deployment.
7.	Model Summary	Layer-wise overview	Displays model architecture including number of layers and trainable parameters.

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d