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

Date	27th January 2025
Team ID	LTVIP2025TMID36392
Project Name	Enchanted Wing: Marvels of Butterfly Species
Maximum Marks	4 Marks

Technical Architecture:

The Butterfly Species Classifier web application is developed using a combination of modern, efficient, and open-source technologies. The frontend is built using **Streamlit**, a Python-based framework that enables quick and interactive web UI development. Users can upload butterfly images and view the predicted species in real time through a simple and intuitive interface.

Example:

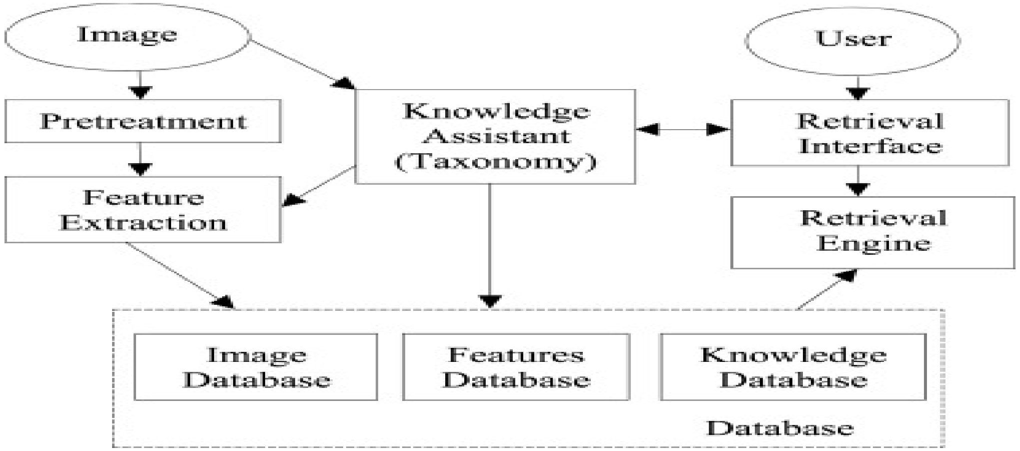

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web interface where user uploads a butterfly image and views prediction	Streamlit (Python-based Web UI)
2	Application Logic-1	Preprocessing the image and making predictions using the trained model	Python, TensorFlow (Keras)
3	Application Logic-2	Decoding predicted labels using saved class indices	Python (pickle, NumPy)
4	Application Logic-3	Displaying prediction result and confidence	Streamlit Components (st.image, st.text)
5	Database	Stores label names and mappings in serialized format	Local `.pkl` file (using Pickle)
6	File Storage	Stores trained model and class indices	Local File System (.keras, .pkl)
7	Machine Learning Model	Used to classify butterfly species from uploaded image	MobileNetV2 (Transfer Learning)
8	Infrastructure	Deployed on local system and optionally via cloud	Local system / Streamlit Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	The project is built using fully open-source tools and libraries.	Streamlit, TensorFlow, Keras, NumPy, Pandas
2	Security Implementations	Since the app doesn't collect personal data, basic file-type validation is used for safe operation.	Streamlit file uploader validation, `.keras` model encapsulation
3	Scalable Architecture	The application uses a modular design, and the model and interface can be easily extended.	Modular Python scripts, scalable Streamlit app structure
4	Availability	The app can be deployed via GitHub and hosted using services like Streamlit Cloud for 24/7 access.	Streamlit Cloud / GitHub Pages (optional)
5	Performance	Lightweight MobileNetV2 model ensures fast response time; handles image input in seconds.	MobileNetV2, PIL (for fast image processing)