**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 26-05-2025 |
| Team ID | LTVIP2025TMID59656 |
| Project Name | Enchanted Wings: Marvels of Butterfly Species |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**Enchanted Wings** is designed with a scalable architecture tailored for real-time image classification using transfer learning. The architecture includes:

* **Presentation Layer (Frontend):**  
  Web/mobile-friendly UI that allows users (researchers, students, citizens) to upload butterfly images, view classification results, and explore educational information.
* **Model & Logic Layer (Backend):**  
  Executes image preprocessing and classification using pre-trained CNN models (e.g., MobileNet, ResNet). Handles species prediction, logging, and result rendering.
* **Data Storage Layer:**  
  Securely stores butterfly images, classification results, dataset metadata, and species-related educational content. Enables retraining and monitoring of model performance.
* **Educational & Monitoring Layer (Optional):**  
  Allows integration of real-time feedback, data visualization, and region-based species tracking for researchers.

**Table-1: Components & Technologies:**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Upload interface for butterfly images, shows predicted species & insights | HTML, CSS, JavaScript. |
|  | |  | | --- | |  |  |  | | --- | | Model Inference API | | Serves classification from trained transfer learning model | Flask / FastAPI, TensorFlow / Keras |
|  | Application Logic | |  | | --- | | Uploads, API endpoints, user handling, educational integration | | Node.js, Express.js |
|  | Admin Panel | |  | | --- | |  |  |  | | --- | | Dataset/image management,  retraining trigger,  model performance viewer | | React.js, Node.js |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Development using modern open-source stacks | Flask |
|  | Scalable Architecture | Layered architecture supporting more species, users, and real-time usage | Restful APIs,  Microservices |

**References:**

[**React.js Documentation**](https://react.dev/)

[**CNN algorithm**](https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/%20s:/www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning)

[**Neural Networks**](https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/)

[**Flask demo**](https://www.youtube.com/watch?v=lj4I_CvBnt0)