**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

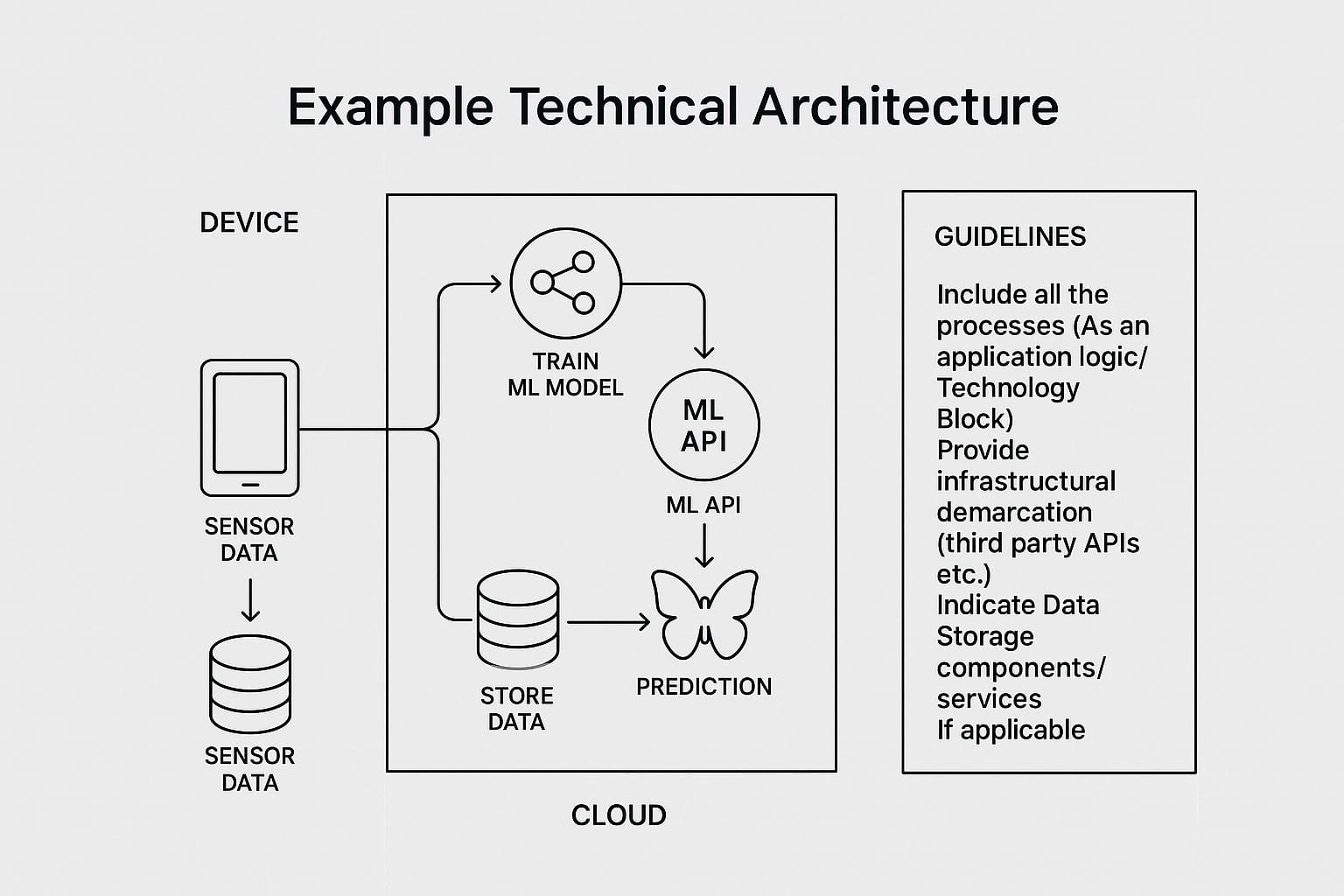
|  |  |
| --- | --- |
| **Date** | 31 January 3035 |
| **Team ID** | LTVIP2025TMID34696 |
| **Project Name** | Enchanted Wings: Marvels Of Butterfly Species |
| **Maximum Marks** | 4 Marks |

**Technical Architecture:**

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

Example: Order processing during pandemics for offline mode

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How users interact with the system through web pages | HTML, CSS, JavaScript, Flask (Python) |
|  | Application Logic-1 | Logic for butterfly image upload, preprocessing, and routing | Python (Flask Framework) |
|  | Application Logic-2 | Transfer Learning Model (VGG16) for butterfly classification | TensorFlow, Keras |
|  | Application Logic-3 | Display species name and confidence, retrieve species info | Python + Flask Backend |
|  | Database | Display species name and confidence, retrieve species info | SQLite / MySQL |
|  | Cloud Database | Stores user data, prediction history, species info | Firebase / Google Cloud SQL |
|  | File Storage | Cloud-hosted database for scalability | Google Drive API / Local Filesystem / Firebase |
|  | External API-1 | Stores uploaded butterfly images and results | Wikipedia API / Custom REST API |
|  | External API-2 | Retrieve butterfly species description from encyclopedia APIs | Google Maps API / GeoTagging API |
|  | Machine Learning Model | Classifies butterflies using deep learningl | VGG16 (Transfer Learning), TensorFlow, Keras |
|  | Infrastructure (Server / Cloud) | Hosting the app locally and optionally on cloud platforms | Localhost, Google Cloud Platform, Render, Heroku |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Web framework and ML libraries used are open-source | Flask, TensorFlow, Keras, Bootstrap |
|  | Security Implementations | Basic authentication and image upload protection; optionally, email OTP | SHA-256, Flask-Login, HTTPS |
|  | Scalable Architecture | Basic authentication and image upload protection; optionally, email OTP | 3-tier Architecture, Flask REST API |
|  | Availability | Can be deployed on cloud platforms for high uptime | Load balancing (Render/Heroku), Auto-scaling |
|  | Performance | Optimized image size (128x128), caching results, quick model inference | Model quantization, ImageDataGenerator, Flask Caching |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)