**1. Architecture Document**

**1.1 Overview**

The **Turbo Engine RUL Prediction System** is a machine learning application that uses **Streamlit** for the user interface and a **TensorFlow** model for predicting the Remaining Useful Life (RUL) of an engine based on operational settings and sensor readings.

**1.2 High-Level Architecture**

The system follows a **single-layer architecture**, where Streamlit serves both as the frontend and backend. Here's how the components are structured:

1. **Frontend and Backend (Streamlit)**:
   * Streamlit handles both the UI and the backend. The frontend collects user inputs and displays the output. It also communicates directly with the TensorFlow model to make predictions.
2. **Machine Learning Model**:
   * A TensorFlow-based deep learning model is used to predict the RUL of an engine.
   * The model is loaded using **TensorFlow**'s load\_model function from a saved .h5 file and requires input data in a specific format for predictions.
3. **Data Flow**:
   * User inputs the operational settings and sensor readings via the UI.
   * The inputs are processed and reshaped within the app itself and passed to the model for prediction.
   * The result is displayed in the app with an explanation.

**1.3 Component Diagram**

+---------------------+ +----------------------+

| User Interface | <---> | ML Model (TensorFlow)|

| (Streamlit) | | (RUL Prediction) |

+---------------------+ +----------------------+

| ^

v |

+---------------------------+ |

| Data Preprocessing & | <--------+

| Reshaping for Prediction |

+---------------------------+

|

v

+-------------------+

| Prediction Result|

+-------------------+

1. **User Interface (Frontend)**:
   * Streamlit UI to collect operational settings and sensor data, and display predictions.
2. **Data Preprocessing**:
   * Data reshaping, padding, and normalization are handled directly within Streamlit before passing to the model.
3. **Machine Learning Model**:
   * A TensorFlow model is used for predicting the engine’s remaining useful life based on the input data.

**1.4 Data Flow Diagram**

+---------------------+ +--------------------+ +-------------------+

| User Inputs (UI) | ----> | Data Preprocessing | ----> | Machine Learning |

| (Operational Setting | | (Reshaping, Padding,| | Model |

| and Sensor Data) | | Normalization) | | (Prediction of RUL)|

+---------------------+ +--------------------+ +-------------------+

|

v

+-------------------+

| Prediction Result|

+-------------------+

1. **User Inputs**: The user inputs operational settings and sensor readings.
2. **Data Preprocessing**: The input data is reshaped, padded, and normalized within the Streamlit app.
3. **Model Prediction**: The preprocessed data is passed to the TensorFlow model to predict the RUL.
4. **Prediction Result**: The predicted RUL is returned to the UI and displayed with an explanation.

**1.5 Technologies Used**

* **Frontend and Backend**: Streamlit (Python)
* **Machine Learning**: TensorFlow
* **Data Processing**: Numpy for reshaping, padding, and transformations