High-Level Design Document for Flight Price Prediction Web Application

1. Architecture Overview

- **Frontend:**
  - **HTML/CSS for the user interface.**
  - **Form handling for user inputs.**
- **Backend:**
  - **Flask for handling HTTP requests and rendering templates.**
  - **Machine Learning model (Using Diffrent Model) for price prediction.**
- **Database:**
  - **MongoDB: Used for storing user input data and possibly storing model predictions or other related data.**

2. Data Flow

1. **User inputs flight details on the web interface.**
2. **Input data is sent to the Flask backend.**
3. **Backend processes data and stores it in MongoDB.**
4. **Data is retrieved from MongoDB, preprocessed, and fed into the ML model.**
5. **The model predicts the price, which is then stored back in MongoDB and displayed to the user.**

3. Module Design

- **Frontend:**
  - **HTML forms for capturing user details.**
- **Backend:**
  - **Flask routes for handling requests.**
  - **Integration with MongoDB using pymongo.**
- **Data Processing Module:**
  - **Preprocessing input data.**
  - **Storing and retrieving data from MongoDB.**

4. Technology Stack

- **Frontend: HTML5, CSS3, Bootstrap.**
- **Backend: Python, Flask, MongoDB (via pymongo).**
- **Machine Learning: Scikit-learn for model training.**

5. Deployment Strategy

- **MongoDB: Hosted on MongoDB Atlas or locally.**
- **Flask App: Deployed on a cloud platform like Heroku or AWS, with the MongoDB connection configured in the environment settings.**

This streamlined HLD captures the essential components and their interactions, focusing on the integration of MongoDB with your Flask-based web application.