Property Valuation – Developer Flow (Simple)

Goal

Build a backend service that predicts property value using minimal inputs and exposes it via a REST API for frontend usage.

System Overview

User → Frontend → REST API → ML Model → JSON Response → Frontend UI

Team Responsibilities

ML Developer

- Train price prediction model
- Save trained model file

Backend Developer

- Build REST API
- Load ML model
- Process request and return response

Frontend Developer

- Collect user inputs
- Call API
- Display results

ML Developer Flow

Step 1: Data Preparation

- Use dataset columns:
- BuiltUpArea_sqft
- BHK
- Latitude
- Longitude
- Price (target)
- Remove nulls and extreme outliers

Step 2: Model Training

- Use RandomForest or XGBoost
- Inputs: area, bhk, latitude, longitude
- Output: property price

Step 3: Save Model

- Export trained model as:
- `property_valuation_model.pkl`

ML work ends here.

Backend Developer Flow

Step 1: API Setup

- Framework: FastAPI
- Endpoint:
- `POST /api/valuation/predict`

Step 2: Load Model

- Load model once during server startup

Step 3: API Input

```
{
  "area_sqft": 1200,
  "bhk": 2,
  "latitude": 12.9716,
  "longitude": 77.5946
}
```

Step 4: Processing Logic

1. Validate input
2. Run model prediction
3. Calculate price per sqft
4. Create price range (±10%)
5. Generate confidence score

Step 5: API Response

```
{
  "valuation": {
    "estimated_price": 8750000,
    "price_per_sqft": 7291
  },
  "price_range": {
    "low": 8300000,
    "high": 9200000
  },
  "location_insights": {
    "city": "Bangalore"
  },
  "confidence_score": 0.86
}
```

Backend work ends here.

Frontend Developer Flow

1. Collect area, BHK, and location from user
2. Send POST request to API
3. Display price, range, and confidence score

Frontend performs no calculations.

Phase-wise Development

Phase 1 (Mandatory)

- Price prediction
- Price range
- Confidence score

Phase 2 (Optional)

- Comparable properties
- Location insights

Summary

- ML builds model
- Backend exposes model as API
- Frontend consumes API
- Clean separation of responsibilities