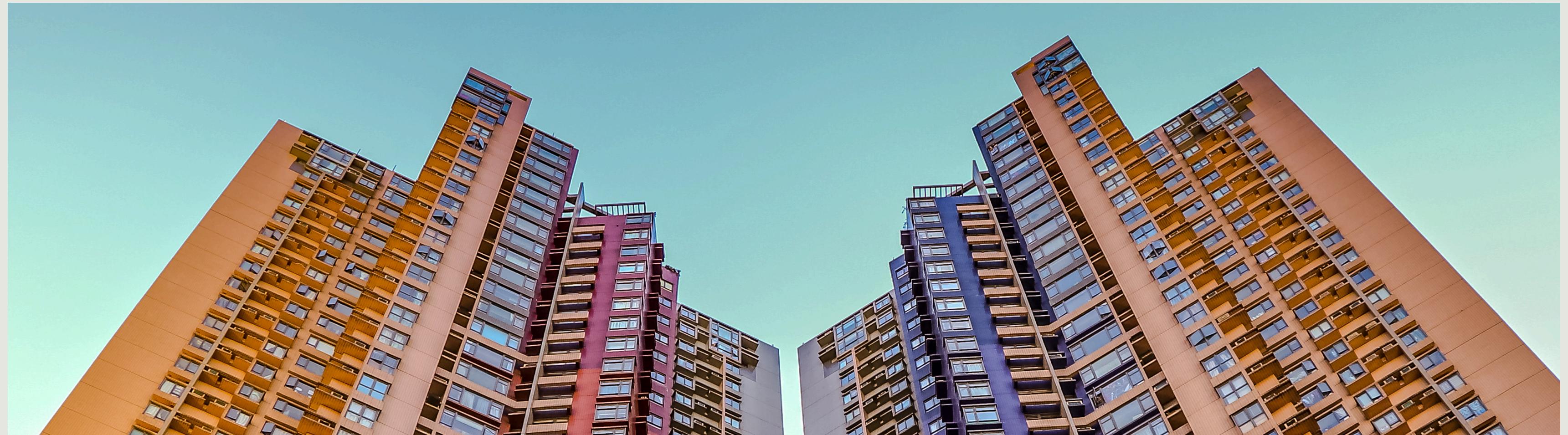
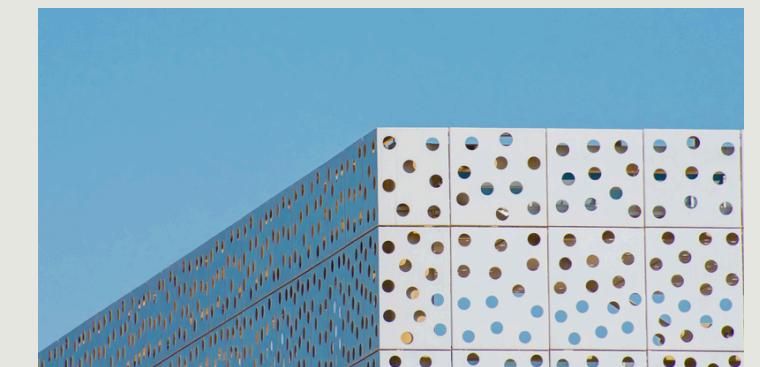


# PROP-AI

AI-Powered Property  
Price Aggregator &  
Prediction Platform



# TOPIC

This project aims to design and implement a web-based property price aggregator platform.

The system provides a modern and user friendly interface, supported by a machine learning model that analyzes property attributes such as location, size, and facilities to generate accurate price estimates. The price recommendation feature helps sellers set competitive prices while increasing market transparency for buyers in assessing the fairness of property prices.



# ROLES AND RESPONSIBILITIES

Software Engineering



**GANENDRA GARDA PRATAMA**

PROJECT LEAD / SYSTEM  
ARCHITECTURE & DATABASE  
COLLECTION

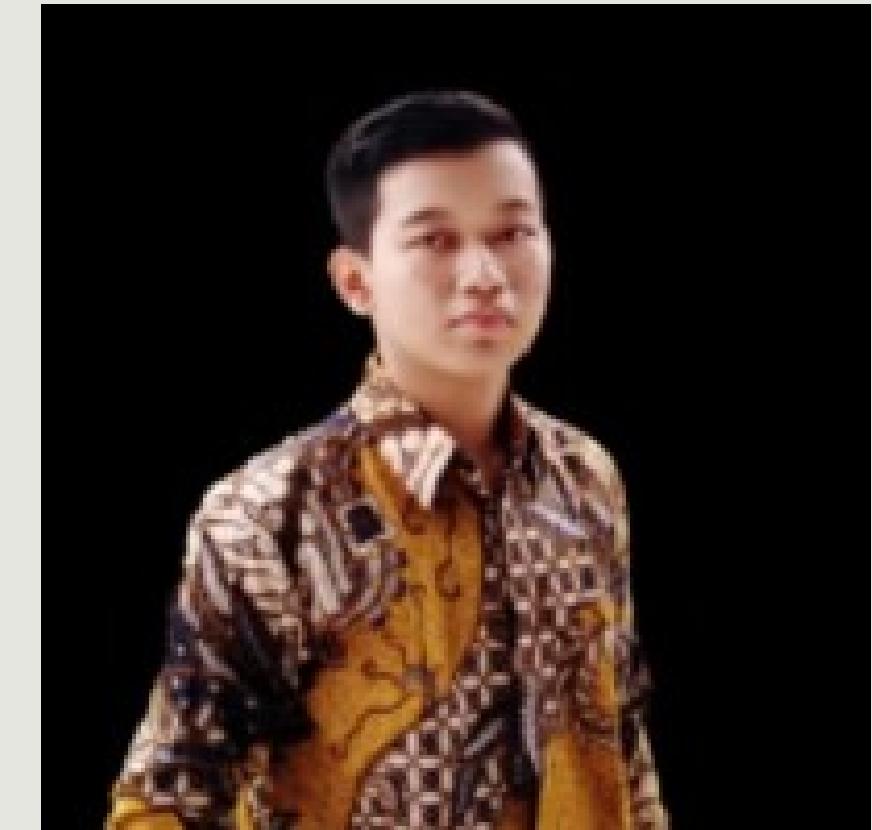
**MUHAMMAD RAIHAN MUSTOFA**

CLIENT APP DEVELOPER



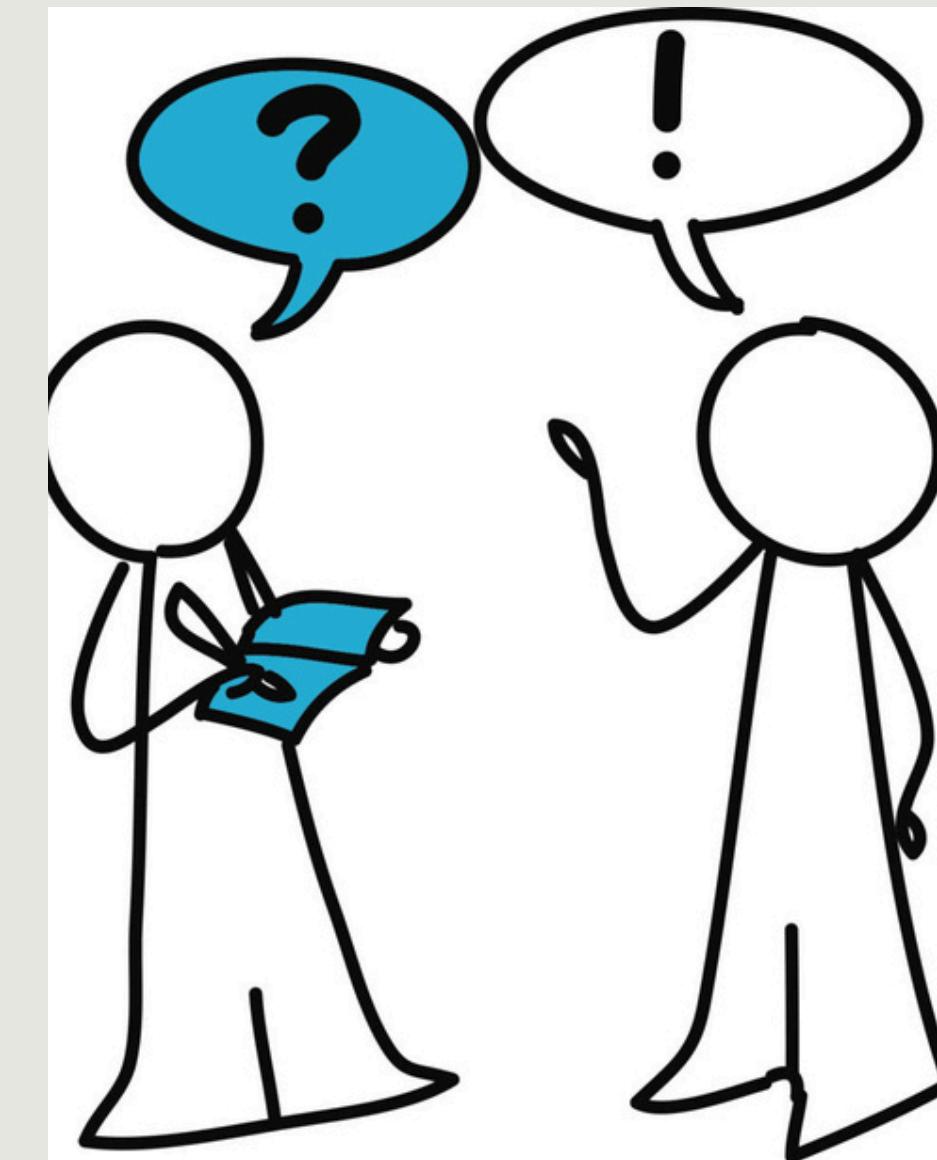
**RAKA ARRAYAN MUTTAQIEN**

AI ENGINEER & DATABASE  
MANAGEMENT



# PROBLEM STATEMENT

- **Lack of Transparency:** Buyers struggle to assess fair property prices due to hidden market data.
- **Pricing Difficulty:** Sellers find it hard to set competitive prices without objective benchmarks.
- **Subjective Valuation:** Reliance on intuition rather than data leads to unfair negotiation and transactions.
- **Market Inefficiency:** The absence of data-driven insights slows down the decision-making process.



# THE SOLUTION

- **Web-Based Aggregator:** A modern, centralized platform for all property needs.
- **Machine Learning Engine:** Analyzing location, size, and facilities to generate accurate estimates.
- **Objective Predictions:** Removing bias to ensure fair pricing for both parties.
- **Market Transparency:** Empowering users with data-driven confidence in their transactions.



# SIMILAR COMPETITORS

Software Engineering



# CORE FEATURE

Software Engineering

- AI powered property price prediction
- Optimal price recommendations for sellers
- Market data & price visualization
- Admin dashboard for content moderation & monitoring
- Secure user authentication & profile management
- Comprehensive property listing details & history



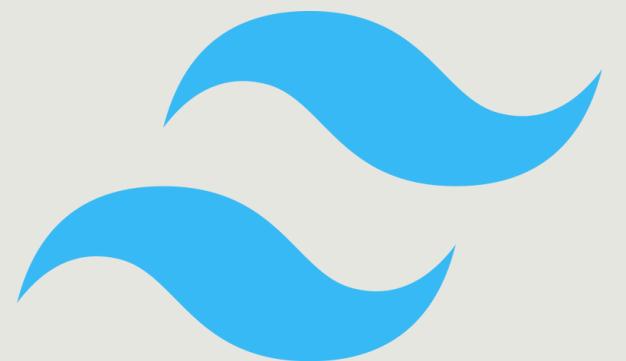
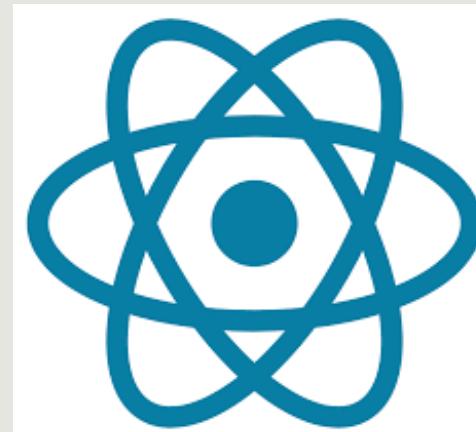
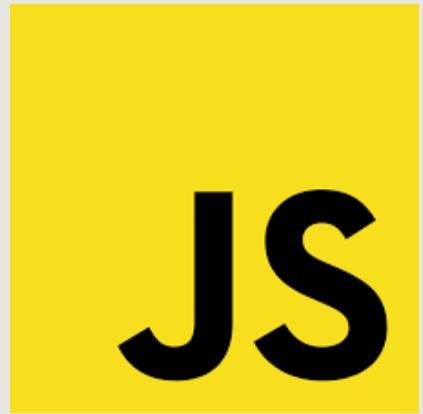
## User

- The user fills in or enters specific data (such as location, land area, property type, and other factors) as input to the system.
- After the data is entered, the system processes the information and provides a land or property price estimate based on the data entered by the user.

## Admin

- The admin is tasked with scraping the latest data and then updating the system database with that information

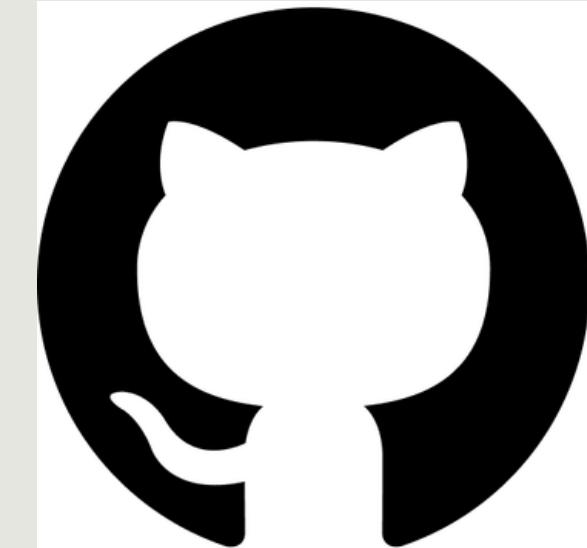
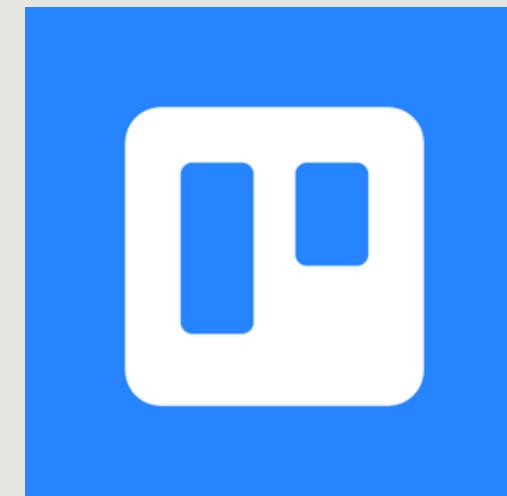
## FRONTEND



## AI



## PROJECT MANAGEMENT



# RISK ANALYSIS

Software Engineering

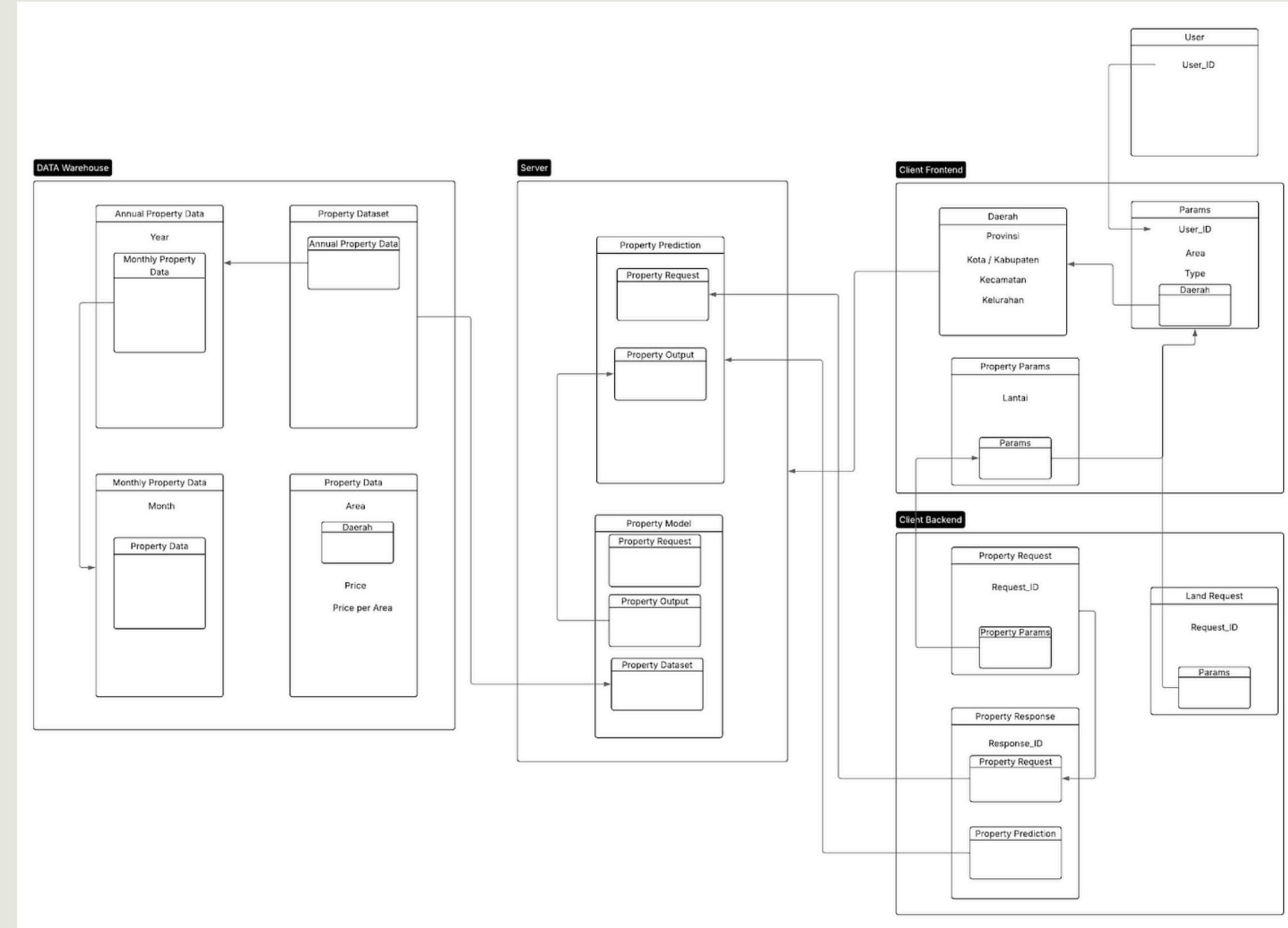
Category	Key Risk	Mitigation
Technical	Technical Integration	Conduct early integration testing, use standardized APIs, and maintain clear documentation.
Data	Data Quality	Perform regular data cleaning and use reliable data sources.
Security	Data Security	Apply encryption, strong authentication, and periodic security audits.
System	System Scalability	Use cloud infrastructure and load balancing to prevent server overload.
Market	Market Acceptance	Conduct promotions, offer user incentives, and improve features based on feedback.

# PROJECT TIMELINE

Software Engineering

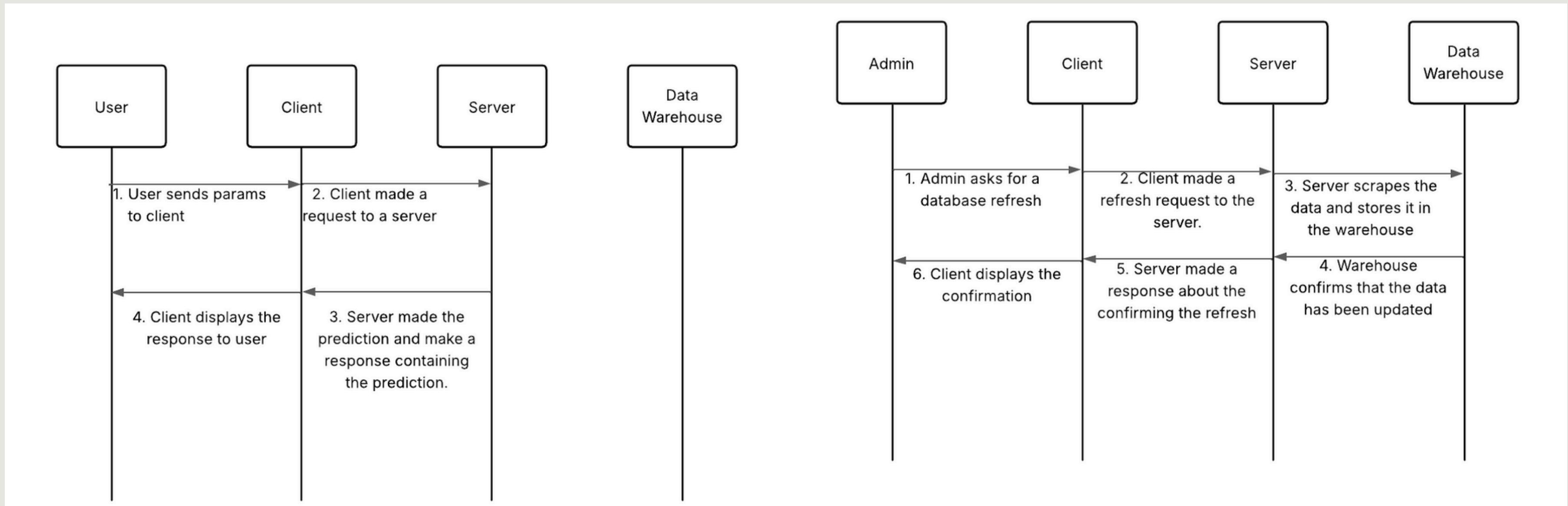
Aspect	September				October				November				December			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
<b>Planning</b>																
Project Goals and Idea	■	■														
Consulting Project	■	■	■	■												
Finalize Project Details		■	■	■	■											
<b>ANALYSIS</b>																
Form the Development Team			■	■												
Assign Team Roles				■	■											
Scheduling					■	■										
<b>DESIGN</b>																
UML Design					■	■										
UI/UX Design						■	■	■								
<b>IMPLEMENTATION</b>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Back-End & Database Development						■	■	■	■	■	■					
Front-End Development						■	■	■	■							
Model						■	■	■	■	■	■					
<b>TESTING</b>																
Integration and Deployment						■	■	■	■	■						
Testing											■	■	■	■	■	
Quality Assurance											■	■	■	■	■	

# CLASS DIAGRAM

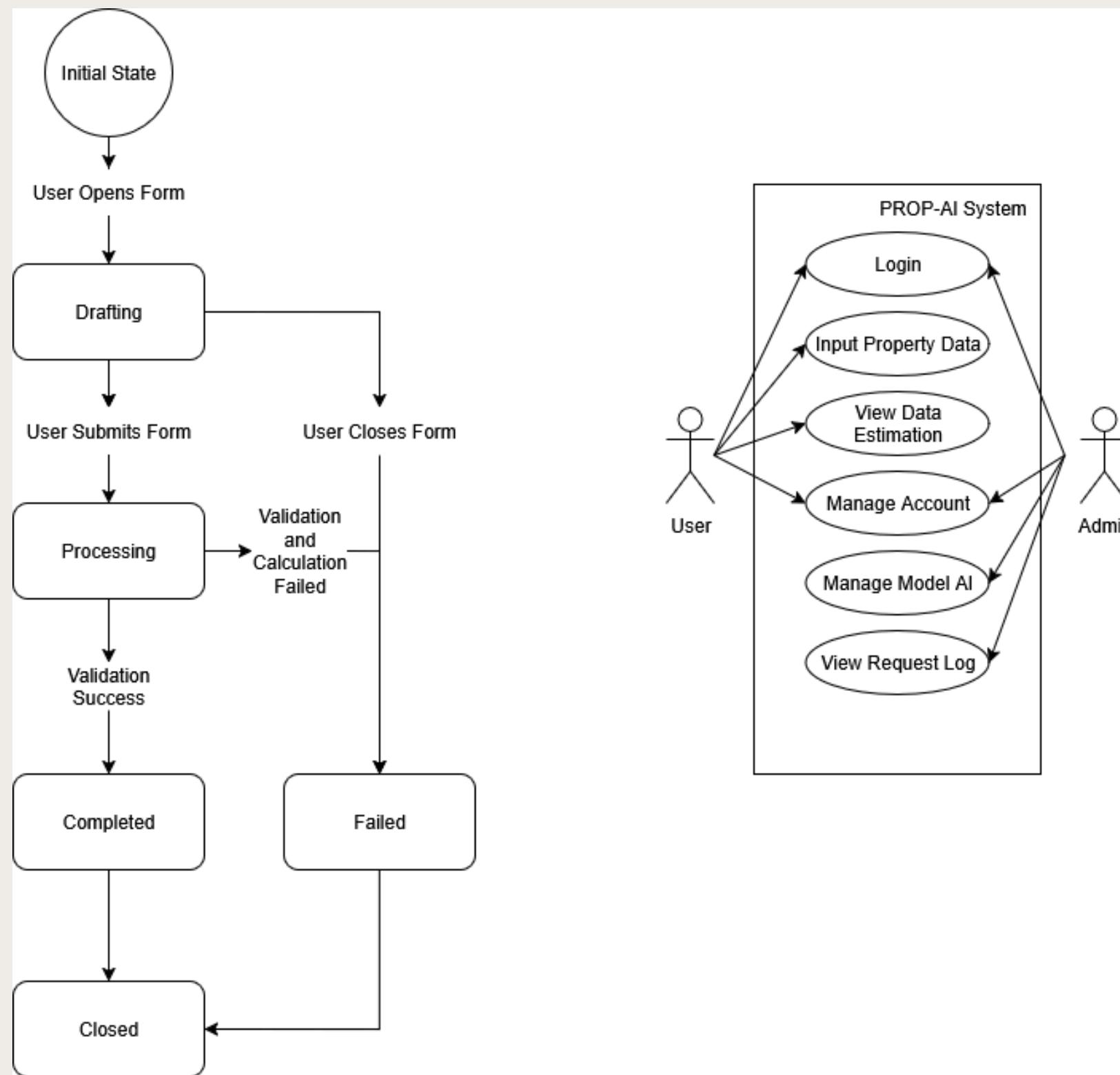


# SEQUENCE DIAGRAM - PREDICTION AND REFRESH

Software Engineering

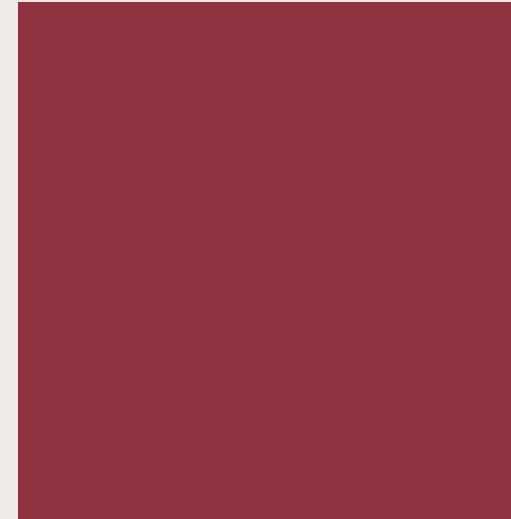
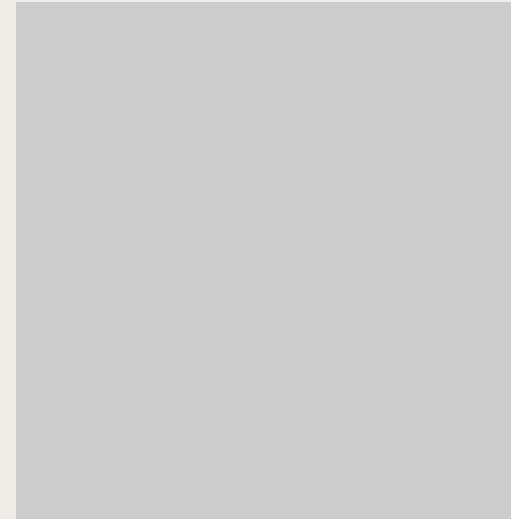
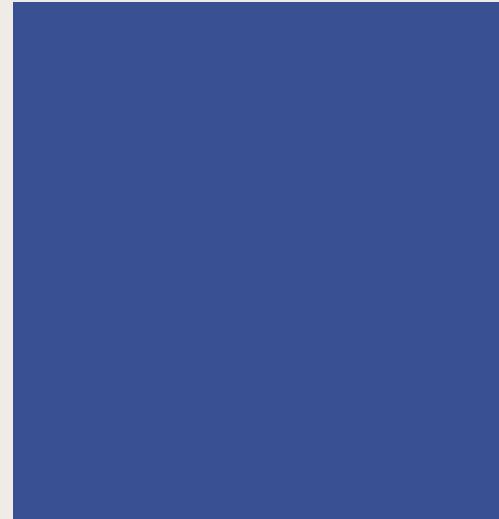


# STATE & USE CASE DIAGRAM



# DESIGN IMPLEMENTATION

Software Engineering



**PROP-AI** Home How It Works Login Sign Up

Location (City/Neighborhood)

Land Size (in meters)

Rooms (Count)

Garage

 Add Other Facilities

Contact Us Privacy Policy

```

from sklearn.ensemble import RandomForestRegressor # model machine Learning berbasis ensemble
from sklearn.model_selection import RandomizedSearchCV # untuk hyperparameter tuning untuk mencari kombinasi parameter t
from sklearn.metrics import mean_absolute_error, r2_score # metrik untuk menilai performa model regresi
# MAE adalah seberapa jauh rata-rata prediksi dari nilai sebenarnya
# R^2 adalah seberapa baik model menjelaskan variasi data

param_dist = {
    'n_estimators': [100, 200, 300, 400], # jumlah pohon yang dibangun
    'max_depth': [5, 10, 15, 20, 25], # kedalaman maksimum tiap pohon
    'min_samples_split': [2, 4, 6, 8], # jumlah minimal sampel untuk memecah node.
    'min_samples_leaf': [1, 2, 4], # jumlah minimal sampel di setiap daun (Leaf)
    'bootstrap': [True, False], # menggunakan sampling acak
    'max_features': ['auto', 'sqrt', 'log2'] # jumlah fitur maksimum yang digunakan untuk membagi node
}

rf_model = RandomForestRegressor(random_state=42) # Membuat base model Random Forest dengan random_state

random_search = RandomizedSearchCV(
    estimator=rf_model, # model yang akan dituning adalah RandomForestRegressor
    param_distributions=param_dist, # daftar parameter yang akan diuji secara acak.
    n_iter=50, # hanya 50 kombinasi acak yang akan diuji
    scoring='r2', # metrik yang digunakan untuk menentukan kombinasi terbaik.
    cv=5, # menggunakan 5-fold cross-validation
    verbose=2, # menampilkan progres proses tuning di terminal.
    random_state=42, # supaya hasil konsisten
    n_jobs=-1 # gunakan semua core CPU
)

random_search.fit(X_train, y_train) # Melatih model RandomForestRegressor menggunakan kombinasi parameter acak dari param

best_rf_model = random_search.best_estimator_ # Menyimpan model Random Forest terbaik hasil tuning

y_train_pred = best_rf_model.predict(X_train) # hasil prediksi pada data training.
y_test_pred = best_rf_model.predict(X_test) # hasil prediksi pada data testing

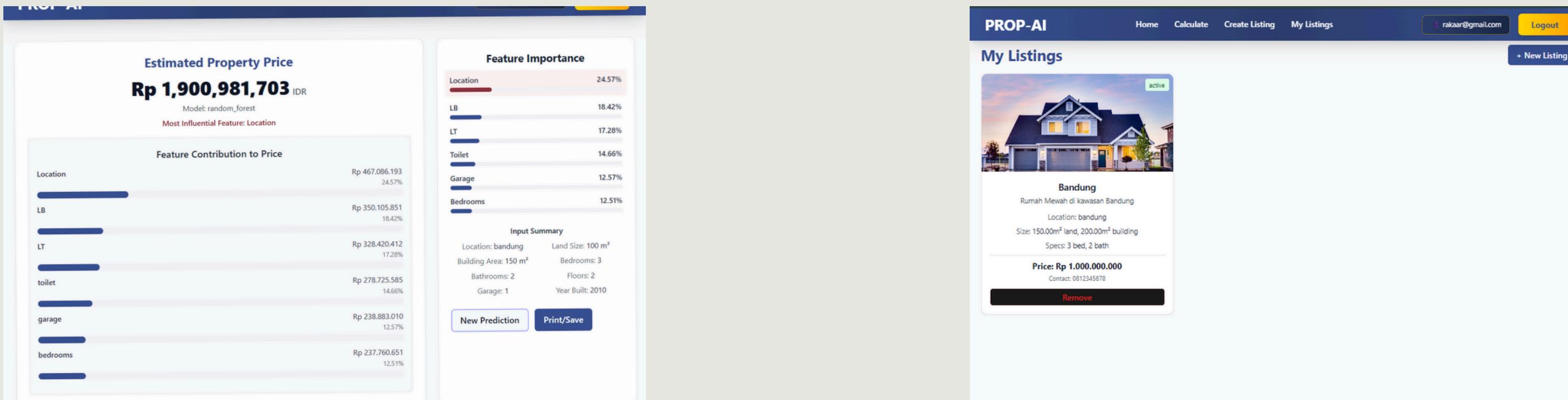
print(f"Train R^2: {r2_score(y_train, y_train_pred):.3f}") # seberapa baik model belajar dari data pelatihan
print(f"Test R^2: {r2_score(y_test, y_test_pred):.3f}") # seberapa baik model memprediksi data baru
print(f"MAE: Rp {mean_absolute_error(y_test, y_test_pred):,.0f}") # rata-rata selisih absolut antara nilai sebenarnya da

```

- Uses a Random Forest Regressor for price prediction.
- Hyperparameters are tuned using RandomizedSearchCV with 5 fold cross validation.
- Selects the best parameter combination based on the R<sup>2</sup> score.
- Applies the optimized model for predictions on training and testing data.
- Evaluates performance using R<sup>2</sup> and MAE.

# KEY FEATURES

Software Engineering



## AI & Pricing

- AI-powered price prediction
- Optimal price recommendations for sellers

## Data & Market

- Comprehensive listing details & history
- Market data & price visualization

# USER WORKFLOW

Software Engineering

**Users must first login or register to access personalized features.**

**Choose Your Action**

- Price Prediction:** Input property details to get an instant AI valuation.
- Market Overview:** Browse current market listing from other web.
- Create Listing:** Sellers can post and manage their own property advertisements.



**PROP-AI**  
FROM DATA TO DECISION

**THANK  
YOU**