# Property Hub

## Senior Full Stack Developer Evaluation

This take-home exercise is designed to evaluate your problem-solving ability, code quality, architecture design, and presentation skills in a realistic, AI-assisted development scenario. You are free to use tools such as GPT, Claude, or Copilot, but your final submission should reflect your own architectural decisions, clean coding practices, and thoughtful UI/UX design.

## Overview

The objective is to build a Property Hub service and user interface that consolidates property details from multiple OTAs (Online Travel Agencies) into a unified, canonical format. You will process JSON datasets for 3 properties from 3 OTAs (Yanolja, otaA, otaY), totalling 9 JSON files. Each JSON will have a different structure.

## Requirements

1. Data Normalization & Canonical JSON Creation:  
 - Extract key property attributes and transform them into a canonical JSON.  
 - Required fields: Name, Address, Description, Nearby Attractions, Images, Facilities.  
 - Add any other fields that enhance the property profile.

2. Data Comparison & Similarity Scoring:  
 - Use Yanolja as the base reference.  
 - Compare Yanolja vs otaA and Yanolja vs otaY.  
 - Compute match percentages for Name, Address, Images, Facilities, and Overall Match.

**NOTE:** Images links for otaA and otaY may be broken and so Image matching should be done last by using Yanolja Images on both sides.

3. Service Layer:  
 - API to fetch consolidated and OTA-specific property data.  
 - Input: Property ID.  
 - Output: Canonical JSON + comparison scores.

4. User Interface:  
 - Consolidated Property View with match percentages.  
 - OTA-specific raw data view.  
 - Creative UI elements to enhance understanding.

## Challenges to Address

* Handling schema differences across OTAs.
* Managing inconsistent or incomplete data.
* Designing effective matching algorithms.
* Presenting comparison results in a user-friendly way.

## Deliverables

1. Backend service (API endpoints, matching logic).
2. Frontend application (consolidated & OTA-specific views).
3. Documentation: Canonical JSON schema, API specification, approach notes.
4. (Optional) Short Loom/video walkthrough.

## Evaluation Rubric

* Architecture & Problem Solving - 30%
* Code Quality & Documentation - 30%
* UI/UX & Presentation - 25%
* Creativity & Extra Features - 15%

## Instructions

Suggested Duration: 48-72 hours  
Submission Format: Zip of Git repository with README  
Include setup steps, assumptions, and architectural decisions.

**Disclaimer**

All OTA names used in this exercise are for simulation purposes only. No affiliation, endorsement, or real data usage is intended.

All data used in this exercise is fictional and for evaluation purposes only. Any resemblance to real properties, OTAs, or data is purely coincidental.