

Part 1:

Task 1.1:

1. Create variables for 2 more properties (property2, property3)
2. Calculate price per m² for all 3

For each of the three properties, we declare variables holding (name, price, area) and calculate price per square meter for each property.

```
func main() {  
    // Property 1  
    var property1Name string = "Saigon Apartment"  
    var property1Price float64 = 2500000000  
    var property1Area float64 = 75.5  
    pricePerM2_1 := property1Price / property1Area  
  
    // Property 2  
    var property2Name string = "Hanoi Condo"  
    var property2Price float64 = 2800000000  
    var property2Area float64 = 90.0  
    pricePerM2_2 := property2Price / property2Area  
  
    // Property 3  
    var property3Name string = "Da Nang Villa"  
    var property3Price float64 = 3200000000  
    var property3Area float64 = 120.0  
    pricePerM2_3 := property3Price / property3Area
```

Prints the price per m² for all three properties.

```
24    // Print property comparison  
25    fmt.Println("== Property Comparison ==")  
26    fmt.Printf("Property 1: %s - %.0f VND/m²\n", property1Name, pricePerM2_1)  
27    fmt.Printf("Property 2: %s - %.0f VND/m²\n", property2Name, pricePerM2_2)  
28    fmt.Printf("Property 3: %s - %.0f VND/m²\n", property3Name, pricePerM2_3)
```

3. Find and print which property has the lowest price per m²

Uses slices to store property names and prices.

```
30    // Created slices to hold names and prices  
31    names := []string{property1Name, property2Name, property3Name}  
32    prices := []float64{pricePerM2_1, pricePerM2_2, pricePerM2_3}
```

Loops through prices to find the cheapest property and prints the name and price of the cheapest property.

```

30 // Created slices to hold names and prices
31 names := []string{property1Name, property2Name, property3Name}
32 prices := []float64{pricePerM2_1, pricePerM2_2, pricePerM2_3}
33
34 // Find the cheapest per m2
35 cheapestName := names[0]
36 cheapestPrice := prices[0]
37
38 for i := 1; i < len(prices); i++ {
39     if prices[i] < cheapestPrice {
40         cheapestPrice = prices[i]
41         cheapestName = names[i]
42     }
43 }
44
45 fmt.Printf("Cheapest per m2: %s at %.0f VND/m2\n", cheapestName, cheapestPrice)
46 }
```

⇒ Result:

```

PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Task1> go run Task1.1.go
--- Property Comparison ---
Property 1: Saigon Apartment - 33112583 VND/m2
Property 2: Hanoi Condo - 31111111 VND/m2
Property 3: Da Nang Villa - 26666667 VND/m2
Cheapest per m2: Da Nang Villa at 26666667 VND/m2
```

Task 1.2:

1. Use this function to categorize ALL your properties

We add Function that assigns a category based on price per m²:

```

5 // Categorize property based on price per m2
6 func categorizeProperty(pricePerM2 float64) string {
7     if pricePerM2 > 50000000 {
8         return "LUXURY"
9     } else if pricePerM2 > 30000000 {
10        return "PREMIUM"
11    } else if pricePerM2 > 20000000 {
12        return "STANDARD"
13    }
14    return "BUDGET"
15 }
```

Calling categorizeProperty function to classify each category.

```

69     category1 := categorizeProperty(pricePerM2_1)
70     category2 := categorizeProperty(pricePerM2_2)
71     category3 := categorizeProperty(pricePerM2_3)
```

2. Count how many properties are in each category

Apply categorization to each property and count how many properties fall into each category.

```
73     luxuryCount := 0
74     premiumCount := 0
75     standardCount := 0
76     budgetCount := 0
77
78     categories := []string{category1, category2, category3}
79     for _, cat := range categories {
80         if cat == "LUXURY" {
81             luxuryCount++
82         } else if cat == "PREMIUM" {
83             premiumCount++
84         } else if cat == "STANDARD" {
85             standardCount++
86         } else if cat == "BUDGET" {
87             budgetCount++
88         }
89     }
90 }
```

3. Create a new function formatPrice() that converts VND to billions/millions

Convert large VND numbers to readable format:

```
17     // Format price to billions (tỷ) or millions (triệu)
18     func formatPrice(price float64) string {
19         if price >= 1000000000 {
20             billions := price / 1000000000
21             return fmt.Sprintf("%.1f tỷ VND", billions)
22         }
23         millions := price / 1000000
24         return fmt.Sprintf("%.0f triệu VND", millions)
25     }
```

Display the result:

```
91     // Display property categories
92     fmt.Println("\n==== Property Categories ===")
93     fmt.Printf("%s: %s (%s)\n", property1Name, category1, formatPrice(property1Price))
94     fmt.Printf("%s: %s (%s)\n", property2Name, category2, formatPrice(property2Price))
95     fmt.Printf("%s: %s (%s)\n", property3Name, category3, formatPrice(property3Price))
96
97     // Display category summary
98     fmt.Println("\nCategory Summary:")
99     fmt.Printf("LUXURY: %d properties\n", luxuryCount)
100    fmt.Printf("PREMIUM: %d properties\n", premiumCount)
101    fmt.Printf("STANDARD: %d properties\n", standardCount)
102    fmt.Printf("BUDGET: %d properties\n", budgetCount)
103 }
```

⇒ Result:

```
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Task1> go run Task1.go
--- Property Comparison ---
Property 1: Saigon Apartment - 33112583 VND/m2
Property 2: Hanoi Condo - 31111111 VND/m2
Property 3: Da Nang Villa - 26666667 VND/m2

Cheapest per m2: Da Nang Villa at 26666667 VND/m2

--- Property Categories ---
Saigon Apartment: PREMIUM (2.5 tỷ VND)
Hanoi Condo: PREMIUM (2.8 tỷ VND)
Da Nang Villa: STANDARD (3.2 tỷ VND)

Category Summary:
LUXURY: 0 properties
PREMIUM: 2 properties
STANDARD: 1 properties
BUDGET: 0 properties
```

Part 2:

Property structure:

```
1 package main
2
3 import (
4     "fmt"
5     "sort"
6 )
7
8 // Property struct to organize data better
9 type Property struct {
10    Name      string
11    Price     float64
12    Area      float64
13    Bedrooms int
14    District  string
15 }
```

Sample properties:

```

// Sample properties
properties := []Property{
    {"Saigon Apartment", 2500000000, 75.5, 2, "District 1"},
    {"HCMC House", 4200000000, 120.0, 3, "District 7"},
    {"Budget Studio", 800000000, 35.0, 1, "Binh Thanh"},
    {"Luxury Penthouse", 5500000000, 150.0, 3, "District 1"},
    {"Cozy Condo", 1800000000, 60.0, 2, "District 7"},
}

fmt.Println("== All Properties ==")
for i, prop := range properties {
    pricePerM2 := prop.Price / prop.Area
    fmt.Printf("%d. %s: %s (%.0f VND/m²)\n",
        i+1, prop.Name, formatPrice(prop.Price), pricePerM2)
}

```

Task 2.1:

1. Function findPropertiesInBudget(properties []Property, maxBudget float64)

Implements a function that filters and returns all properties with prices less than or equal to a given maximum budget by iterating over the list and checking each price:

```

17 // Find properties within maxBudget
18 func findPropertiesInBudget(properties []Property, maxBudget float64) []Property {
19     var result []Property
20     for _, prop := range properties {
21         if prop.Price <= maxBudget {
22             result = append(result, prop)
23         }
24     }
25     return result
26 }
```

2. Function findPropertiesByBedrooms(properties []Property, bedrooms int)

Function that finds and returns properties matching an exact number of bedrooms by looping through all properties and comparing the bedroom count:

```

28 // Find properties by number of bedrooms
29 func findPropertiesByBedrooms(properties []Property, bedrooms int) []Property {
30     var result []Property
31     for _, prop := range properties {
32         if prop.Bedrooms == bedrooms {
33             result = append(result, prop)
34         }
35     }
36     return result
37 }
```

3. Test both functions with your data

```
// TASK 2.1: Search functions
// Find properties under budget
budget := 3000000000.0
affordable := findPropertiesInBudget(properties, budget)
fmt.Printf("\n==== Properties under %s ====\n", formatPrice(budget))
for _, prop := range affordable {
    fmt.Printf("- %s: %s\n", prop.Name, formatPrice(prop.Price))
}

// Find properties by bedrooms
bedroomSearch := 2
byBedrooms := findPropertiesByBedrooms(properties, bedroomSearch)
fmt.Printf("\n==== Properties with %d bedrooms ====\n", bedroomSearch)
for _, prop := range byBedrooms {
    fmt.Printf("- %s: %s\n", prop.Name, formatPrice(prop.Price))
}
```

Result:

```
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part2> go run Task2.go

Properties under 3000000000 VND:
- Saigon Apartment: 2500000000 VND
- Budget Studio: 800000000 VND
- Cozy Condo: 1800000000 VND

Properties with 2 bedrooms:
- Saigon Apartment: 2500000000 VND
- Cozy Condo: 1800000000 VND
```

Defines a helper function to format property prices into readable strings, converting large values into billions or millions of Vietnamese Dong:

```
39 // Format price for display
40 func formatPrice(price float64) string {
41     if price >= 1000000000 {
42         billions := price / 1000000000
43         return fmt.Sprintf("%.1f tỷ VND", billions)
44     }
45     millions := price / 1000000
46     return fmt.Sprintf("%.0f triệu VND", millions)
47 }
```

Task 2.2:

1. Use the analyzeByDistrict function

Groups properties by their district using a map where each key is a district name and the value is a list of properties in that district.

```
49 // Group properties by district
50 func analyzeByDistrict(properties []Property) map[string][]Property {
51     districtMap := make(map[string][]Property)
52     for _, prop := range properties {
53         districtMap[prop.District] = append(districtMap[prop.District], prop)
54     }
55     return districtMap
56 }
```

2. Create calculateDistrictStats() function that returns: - Average price per district - Property count per district - Most expensive property in each district

Introduces a DistrictStats struct to hold statistics:

```
58 // District statistics struct
59 type DistrictStats struct {
60     District      string
61     PropertyCount int
62     AveragePrice  float64
63     MostExpensive Property
64 }
```

Calculates these statistics by iterating over each district's properties, summing prices, finding the highest price, computing the average, and returning the results as a slice of DistrictStats:

```
66 // Calculate district statistics
67 func calculateDistrictStats(districtMap map[string][]Property) []DistrictStats {
68     var stats []DistrictStats
69
70     for district, props := range districtMap {
71         totalPrice := 0.0
72         mostExpensive := props[0]
73
74         for _, prop := range props {
75             totalPrice += prop.Price
76             if prop.Price > mostExpensive.Price {
77                 mostExpensive = prop
78             }
79         }
80
81         avgPrice := totalPrice / float64(len(props))
82
83         stats = append(stats, DistrictStats{
84             District:      district,
85             PropertyCount: len(props),
86             AveragePrice:  avgPrice,
87             MostExpensive: mostExpensive,
88         })
89     }
90
91     return stats
92 }
```

3. Display results sorted by average price (highest first)

```
// TASK 2.2: District analysis
// District analysis
districtMap := analyzeByDistrict(properties)
stats := calculateDistrictStats(districtMap)

fmt.Println("\n==== District Analysis ===")
for _, stat := range stats {
    fmt.Printf("%s: %d properties, Avg: %s, Most expensive: %s\n",
        stat.District,
        stat.PropertyCount,
        formatPrice(stat.AveragePrice),
        stat.MostExpensive.Name)
}

// Sort districts by average price descending
sort.Slice(stats, func(i, j int) bool {
    return stats[i].AveragePrice > stats[j].AveragePrice
})

fmt.Println("\n==== Ranking by Average Price ===")
for i, stat := range stats {
    fmt.Printf("%d. %s: %s\n", i+1, stat.District, formatPrice(stat.AveragePrice))
}
```

⇒ Final result:

```
PS C:\Users\VUTHANHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part2> go run Task2.go
--- All Properties ---
1. Saigon Apartment: 2.5 tỷ VND (33112583 VND/m2)
2. HCMC House: 4.2 tỷ VND (35000000 VND/m2)
3. Budget Studio: 800 triệu VND (22857143 VND/m2)
4. Luxury Penthouse: 5.5 tỷ VND (36666667 VND/m2)
5. Cozy Condo: 1.8 tỷ VND (30000000 VND/m2)

--- Properties under 3.0 tỷ VND ---
- Saigon Apartment: 2.5 tỷ VND
- Budget Studio: 800 triệu VND
- Cozy Condo: 1.8 tỷ VND

--- Properties with 2 bedrooms ---
- Saigon Apartment: 2.5 tỷ VND
- Cozy Condo: 1.8 tỷ VND

--- District Analysis ---
District 1: 2 properties, Avg: 4.0 tỷ VND, Most expensive: Luxury Penthouse
District 7: 2 properties, Avg: 3.0 tỷ VND, Most expensive: HCMC House
Binh Thanh: 1 properties, Avg: 800 triệu VND, Most expensive: Budget Studio

--- Ranking by Average Price ---
1. District 1: 4.0 tỷ VND
2. District 7: 3.0 tỷ VND
3. Binh Thanh: 800 triệu VND
```

Part 3:

Task 3.1:

```

import [
    "fmt"
    "math"
]

// Property struct holds real estate info
type Property struct {
    Name      string
    Price     float64
    Area      float64
    Bedrooms int
    District  string
}

```

Property Methods:

```

func (p Property) PricePerM2() float64 {
    if p.Area == 0 {
        return 0
    }
    return p.Price / p.Area
}

func (p Property) IsAffordable(budget float64) bool {
    return p.Price <= budget
}

func formatPrice(price float64) string {
    if price >= 1000000000 {
        billions := price / 1000000000
        return fmt.Sprintf("%.1f tỷ VND", billions)
    }
    millions := price / 1000000
    return fmt.Sprintf("%.0f triệu VND", millions)
}

```

```

// --- Main function ---

func main() {
    // Sample properties
    properties := []Property{
        {"Saigon Apartment", 2500000000, 75.5, 2, "District 1"},
        {"HCMC House", 4200000000, 120.0, 3, "District 7"},
        {"Budget Studio", 80000000, 35.0, 1, "Binh Thanh"},
    }

    // Monthly rents for each property
    monthlyRents := []float64{25000000, 35000000, 12000000}
}

```

The CalculateROI method computes the annual return on investment as a percentage based on monthly rent and property price:

```

func (p Property) CalculateROI(monthlyRent float64) float64 {
    annualRent := monthlyRent * 12
    roi := (annualRent / p.Price) * 100
    return roi
}

```

The InvestmentGradeWithRent method categorizes properties into investment quality tiers by comparing the ROI against predefined thresholds:

```

func (p Property) InvestmentGradeWithRent(monthlyRent float64) string {
    roi := p.CalculateROI(monthlyRent)

    if roi > 8 {
        return "EXCELLENT"
    } else if roi >= 5 {
        return "GOOD"
    } else if roi >= 3 {
        return "FAIR"
    }
    return "POOR"
}

```

The findBestInvestment function iterates over properties and their rents to identify and return the property with the highest ROI:

```

func findBestInvestment(properties []Property, rents []float64) (Property, float64) {
    bestProperty := properties[0]
    bestROI := properties[0].CalculateROI(rents[0])

    for i := 1; i < len(properties); i++ {
        roi := properties[i].CalculateROI(rents[i])
        if roi > bestROI {
            bestROI = roi
            bestProperty = properties[i]
        }
    }

    return bestProperty, bestROI
}

```

```

// TASK 3.1: Investment Analysis
fmt.Println("== Investment Analysis ==")
for i, prop := range properties {
    roi := prop.CalculateROI(monthlyRents[i])
    grade := prop.InvestmentGradeWithRent(monthlyRents[i])
    fmt.Printf("%s: ROI %.1f%% per year - %s\n", prop.Name, roi, grade)
}

// Find best investment
bestProp, bestROI := findBestInvestment(properties, monthlyRents)
fmt.Printf("\nBest Investment: %s (%.1f%% ROI)\n", bestProp.Name, bestROI)

```

Result:

```
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part3> go run Task3.go
==== Investment Analysis ====
Saigon Apartment: ROI 12.0% per year - EXCELLENT
HCMC House: ROI 10.0% per year - EXCELLENT
Budget Studio: ROI 18.0% per year - EXCELLENT

Best Investment: Budget Studio (18.0% ROI)
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part3>
```

Task 3.2

The LoanInfo struct holds detailed loan results including loan amount, monthly payment, and total interest.

```
type LoanInfo struct {
    LoanAmount      float64
    MonthlyPayment float64
    TotalInterest   float64
}
```

The calculateMonthlyPayment function uses the mortgage formula to compute monthly loan payments considering interest rate and loan duration.

```
func calculateMonthlyPayment(loanAmount, annualRate float64, years int) float64 {
    monthlyRate := annualRate / 100 / 12
    numPayments := float64(years * 12)

    if annualRate == 0 {
        return loanAmount / numPayments
    }

    // Mortgage formula: M = P * [r(1+r)^n] / [(1+r)^n - 1]
    numerator := loanAmount * monthlyRate * math.Pow(1+monthlyRate, numPayments)
    denominator := math.Pow(1+monthlyRate, numPayments) - 1
    return numerator / denominator
}
```

The CalculateLoan method calculates a property's loan amount after down payment, monthly payments, and total interest over the loan period.

```

func (p Property) CalculateLoan(downPaymentPercent, interestRate float64, years int) LoanInfo {
    downPayment := p.Price * (downPaymentPercent / 100)
    loanAmount := p.Price - downPayment

    monthlyPayment := calculateMonthlyPayment(loanAmount, interestRate, years)

    totalPaid := monthlyPayment * float64(years*12)
    totalInterest := totalPaid - loanAmount

    return LoanInfo{
        LoanAmount:     loanAmount,
        MonthlyPayment: monthlyPayment,
        TotalInterest: totalInterest,
    }
}

```

```

// TASK 3.2: Loan Analysis
fmt.Println("\n==== Loan Analysis ===")
downPayment := 20.0 // 20%
interestRate := 8.5 // 8.5%
loanYears := 20      // 20 years

for _, prop := range properties {
    loanInfo := prop.CalculateLoan(downPayment, interestRate, loanYears)

    fmt.Printf("%s:\n", prop.Name)
    fmt.Printf("  Loan Amount: %s (%.0f%% of price)\n",
              formatPrice(loanInfo.LoanAmount), 100-downPayment)
    fmt.Printf("  Monthly Payment: %s\n", formatPrice(loanInfo.MonthlyPayment))
    fmt.Printf("  Total Interest: %s over %d years\n\n",
              formatPrice(loanInfo.TotalInterest), loanYears)
}

```

⇒ Final result:

```

● PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part3> go run Task3.go
--- Investment Analysis ---
Saigon Apartment: ROI 12.0% per year - EXCELLENT
HCMC House: ROI 10.0% per year - EXCELLENT
Budget Studio: ROI 18.0% per year - EXCELLENT

Best Investment: Budget Studio (18.0% ROI)

--- Loan Analysis ---
Saigon Apartment:
  Loan Amount: 2.0 tỷ VND (80% of price)
  Monthly Payment: 17 triệu VND
  Total Interest: 2.2 tỷ VND over 20 years

HCMC House:
  Loan Amount: 3.4 tỷ VND (80% of price)
  Monthly Payment: 29 triệu VND
  Total Interest: 3.6 tỷ VND over 20 years

Budget Studio:
  Loan Amount: 640 triệu VND (80% of price)
  Monthly Payment: 6 triệu VND
  Total Interest: 693 triệu VND over 20 years

```

Part 4:

⇒ Final result:

```
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part4> go run .
==== Smart Property Recommendations ===

Saigon Apartment (75.5 m2, District 1)
Price: 2.5 tỷ VND | Price/m2: 33 triệu VND
Recommendation: 🔥 BUY NOW - Highly recommended
Details: Bonuses: 3 [Premium location Optimal size Excellent ROI] | Warnings: 0 []

HCMC House (120.0 m2, District 7)
Price: 4.2 tỷ VND | Price/m2: 35 triệu VND
Recommendation: ✅ GOOD BUY - Recommended
Details: Bonuses: 2 [Premium location Excellent ROI] | Warnings: 1 [Large size]

Budget Studio (35.0 m2, Binh Thanh)
Price: 800 triệu VND | Price/m2: 23 triệu VND
Recommendation: ✅ GOOD BUY - Recommended
Details: Bonuses: 2 [Good value per m2 Excellent ROI] | Warnings: 1 [Small size]

Cozy Condo (65.0 m2, District 2)
Price: 1.8 tỷ VND | Price/m2: 28 triệu VND
Recommendation: 🔥 BUY NOW - Highly recommended
Details: Bonuses: 3 [Premium location Optimal size Excellent ROI] | Warnings: 0 []

==== Portfolio Optimization ===
Budget: 8.0 tỷ VND

Selected Properties:
1. Budget Studio: 800 triệu VND (ROI: 18.0%)
2. Saigon Apartment: 2.5 tỷ VND (ROI: 12.0%)
3. Cozy Condo: 1.8 tỷ VND (ROI: 12.0%)

Total Invested: 5.1 tỷ VND
Remaining Budget: 2.9 tỷ VND
Portfolio Average ROI: 14.0%
```

Part 5:

Property Analyzer Menu:

```
PS C:\Users\VUTHANHHUNG\Desktop\Netcen Pro\VuThanhNhan - ITITIU21267 - Lab1\Part5> go run .
==== Property Analyzer Menu ===
1. View all properties
2. Search by budget
3. Investment analysis
4. Loan calculator
5. Get recommendations
6. Optimize portfolio
0. Exit
```

Choose option: 1

==== All Properties ===

1. Saigon Apartment

Price: 2.5 tỷ VND | Area: 75.5 m² | Bedrooms: 2
District: District 1 | Price/m²: 33 triệu VND

2. HCMC House

Price: 4.2 tỷ VND | Area: 120.0 m² | Bedrooms: 3
District: District 7 | Price/m²: 35 triệu VND

3. Budget Studio

Price: 800 triệu VND | Area: 35.0 m² | Bedrooms: 1
District: Binh Thanh | Price/m²: 23 triệu VND

4. Cozy Condo

Price: 1.8 tỷ VND | Area: 65.0 m² | Bedrooms: 2
District: District 2 | Price/m²: 28 triệu VND

5. Luxury Penthouse

Price: 5.5 tỷ VND | Area: 150.0 m² | Bedrooms: 3
District: District 1 | Price/m²: 37 triệu VND

Choose option: 2

Enter your budget (VND): 2000000000

==== Properties under 2.0 tỷ VND ===

✓ Budget Studio: 800 triệu VND

✓ Cozy Condo: 1.8 tỷ VND

Choose option: 3

==== Investment Analysis ===

Saigon Apartment:

Monthly Rent: 25 triệu VND
ROI: 12.0% per year - EXCELLENT

HCMC House:

Monthly Rent: 35 triệu VND
ROI: 10.0% per year - EXCELLENT

Budget Studio:

Monthly Rent: 12 triệu VND
ROI: 18.0% per year - EXCELLENT

Cozy Condo:

Monthly Rent: 18 triệu VND
ROI: 12.0% per year - EXCELLENT

Luxury Penthouse:

Monthly Rent: 45 triệu VND
ROI: 9.8% per year - EXCELLENT

🏆 Best Investment: Budget Studio (18.0% ROI)

Press Enter to continue... █

Choose option: 4

==== Loan Calculator ====

Down payment percentage (e.g., 20 for 20%): 20

Annual interest rate (e.g., 8.5 for 8.5%): 8.5

Loan term in years: 20

--- Loan Details ---

Saigon Apartment:

Property Price: 2.5 tỷ VND

Loan Amount: 2.0 tỷ VND (80% of price)

Monthly Payment: 17 triệu VND

Total Interest: 2.2 tỷ VND over 20 years

HCMC House:

Property Price: 4.2 tỷ VND

Loan Amount: 3.4 tỷ VND (80% of price)

Monthly Payment: 29 triệu VND

Total Interest: 3.6 tỷ VND over 20 years

Budget Studio:

Property Price: 800 triệu VND

Loan Amount: 640 triệu VND (80% of price)

Monthly Payment: 6 triệu VND

Total Interest: 693 triệu VND over 20 years

Cozy Condo:

Property Price: 1.8 tỷ VND

Loan Amount: 1.4 tỷ VND (80% of price)

Monthly Payment: 12 triệu VND

Total Interest: 1.6 tỷ VND over 20 years

Luxury Penthouse:

Property Price: 5.5 tỷ VND

Loan Amount: 4.4 tỷ VND (80% of price)

Monthly Payment: 38 triệu VND

Total Interest: 4.8 tỷ VND over 20 years

Choose option: 5

Enter your budget (VND): 3000000000

Enter max monthly payment you can afford (VND): 25000000

==== Property Recommendations ===

Saigon Apartment (75.5 m², District 1)

Price: 2.5 tỷ VND | Price/m²: 33 triệu VND

Estimated Rent: 25 triệu VND/month

Recommendation: 🔥 BUY NOW - Highly recommended

Details: Bonuses: 3 [Premium location Optimal size Excellent ROI] | Warnings: 0 []

HCMC House (120.0 m², District 7)

Price: 4.2 tỷ VND | Price/m²: 35 triệu VND

Estimated Rent: 35 triệu VND/month

Recommendation: ✗ SKIP - Over budget

Details: Cannot afford this property

Budget Studio (35.0 m², Binh Thanh)

Price: 800 triệu VND | Price/m²: 23 triệu VND

Estimated Rent: 12 triệu VND/month

Recommendation: ✓ GOOD BUY - Recommended

Details: Bonuses: 2 [Good value per m² Excellent ROI] | Warnings: 1 [Small size]

Cozy Condo (65.0 m², District 2)

Price: 1.8 tỷ VND | Price/m²: 28 triệu VND

Estimated Rent: 18 triệu VND/month

Recommendation: 🔥 BUY NOW - Highly recommended

Details: Bonuses: 3 [Premium location Optimal size Excellent ROI] | Warnings: 0 []

Luxury Penthouse (150.0 m², District 1)

Price: 5.5 tỷ VND | Price/m²: 37 triệu VND

Estimated Rent: 45 triệu VND/month

Recommendation: ✗ SKIP - Over budget

Details: Cannot afford this property

Choose option: 6

Enter your total investment budget (VND): 8000000000

==== Portfolio Optimization ===

Budget: 8.0 tỷ VND

Selected Properties:

1. Budget Studio: 800 triệu VND (ROI: 18.0%)
2. Saigon Apartment: 2.5 tỷ VND (ROI: 12.0%)
3. Cozy Condo: 1.8 tỷ VND (ROI: 12.0%)

Total Invested: 5.1 tỷ VND

Remaining Budget: 2.9 tỷ VND

Portfolio Average ROI: 14.0%

Choose option: 0

👋 Thank you for using Property Analyzer!

Goodbye!