T2India Routing Input Handling Design

System Architecture for Processing Client Input

📥 INPUT PROCESSING FLOW

```
Client Input: "Goa Delhi Kolkata for 12 days"
                  ↓
          [Input Parser & Analyzer]
                  ↓
    ┌─────────────────────────────┐
    │   EXTRACTED INFORMATION      │
    │ • Destinations: [Goa, Delhi, │
    │   Kolkata]                   │
    │ • Duration: 12 days          │
    │ • Type: Custom Route         │
    └─────────────────────────────┘

                  ↓
        [Route Optimization Engine]
                  ↓
    ┌─────────────────────────────┐
    │   OPTIMIZED ROUTING          │
    │ • Best Sequence: Delhi →     │
    │   Goa → Kolkata → Delhi      │
    │ • Travel Logic: Geographical │
    │   optimization               │
    │ • Entry/Exit: Delhi (flights)│
    └─────────────────────────────┘

                  ↓
        [Duration Split Calculator]
                  ↓
    ┌─────────────────────────────┐
    │   TIME ALLOCATION            │
    │ • Main Circuit: 6-7 days     │
    │ • Extensions: 5-6 days       │
    │ • Total: 12 days             │
    └─────────────────────────────┘

                  ↓
      [Extension Suggestion Engine]
                  ↓
    ┌─────────────────────────────┐
    │   THEMED OPTIONS             │
    │ • From Kolkata: Darjeeling,  │
    │   Sikkim, Puri               │
    │ • From Goa: Hampi, Bangalore │
    │ • From Delhi: Rishikesh      │
    └─────────────────────────────┘

                  ↓
       [Connectivity Data Integration]
                  ↓
```

```
┌─────────────────────────────┐
│   REAL TRANSPORT DATA       │
│ • Kolkata-Darjeeling: Train │
│   11h 48m, ₹3K-11K          │
│ • Goa-Hampi: Road 5h, ₹350  │
│ • Delhi-Rishikesh: Road 6h  │
└─────────────────────────────┘
              ↓
      [Client Choice Presentation]
```

🎯 DETAILED ROUTING LOGIC

Step 1: Input Analysis

```python
def extract_destinations(user_input):
    input_lower = user_input.lower()

    # Pattern Recognition
    destinations = []
    duration = None

    # Extract cities
    city_patterns = {
        "goa": "Goa",
        "delhi": "Delhi",
        "kolkata": "Kolkata"
    }

    # Extract duration
    duration_match = re.search(r'(\d+)\s*days?', input_lower)
    if duration_match:
        duration = int(duration_match.group(1))

    return destinations, duration
```

Step 2: Route Optimization

```python
def optimize_route(destinations):
    # Geographical Logic
    region_priority = {
        "Delhi": 1,    # North - good entry point
        "Goa": 2,      # West - coastal
        "Kolkata": 3   # East - good exit point
    }

    # Sort by geographical logic
    optimized = sorted(destinations,
                    key=lambda x: region_priority.get(x, 999))

    # Create circular route
    return optimized + [optimized[0]]  # Return to start
```

Step 3: Duration Allocation

```python
def allocate_duration(total_days, destinations):
    base_days = len(destinations) * 2  # 2 days per city
    remaining_days = total_days - base_days

    return {
        "main_circuit": base_days,
        "extensions": remaining_days,
        "flexibility": remaining_days // len(destinations)
    }
```

🎨 USER INTERFACE DESIGN

Phase 1: Input Collection

```
┌─────────────────────────────────────────┐
│  T2India Intelligent Route Planner    │
├─────────────────────────────────────────┤
│                                         │
│  Tell us your travel preferences:       │
│                                        │
│   ┌───────────────────────────────────┐ │
│   │ "Goa Delhi Kolkata for 12 days"   │ │
│   └───────────────────────────────────┘ │
│                                         │
│  [Generate Smart Suggestions] 🚀        │
│                                         │
└─────────────────────────────────────────┘
```

Phase 2: Route Presentation

```
┌─────────────────────────────────────────┐
│  🎯 Optimized Route Suggestion        │
├─────────────────────────────────────────┤
│                                         │
│  📍 Delhi → Goa → Kolkata → Delhi       │
│                                         │
│  ⏱️  Main Circuit: 6-7 days             │
│  🎨 Extensions: 5-6 days                │
│  📅 Total Duration: 12 days             │
│                                         │
│  ✈  Entry/Exit: Delhi (International)   │
│                                         │
└─────────────────────────────────────────┘
```

Phase 3: Extension Options

```
┌─────────────────────────────────────────┐
│ 🌟 Choose Your Extension Theme           │
├─────────────────────────────────────────┤
│                                          │
│ From Kolkata (3-4 days):            │    │
│ ○ 🏔️  Mountain & Tea Gardens        │    │
│    Darjeeling, Sikkim               │    │
│    Train: 11h 48m, ₹3K-11K          │    │
│                                          │
│ ○ 🕉️  Religious & Spiritual          │    │
│    Puri (Jagannath Temple)          │    │
│    Road: 6h, ₹500                   │    │
│                                          │
│ From Goa (2-3 days):                │    │
│ ○ 🏛️  UNESCO Heritage                │    │
│    Hampi (Ancient Ruins)            │    │
│    Road: 5h, ₹350                   │    │
│                                          │
│ From Delhi (3-4 days):              │    │
│ ○ 🧘 Yoga & Spirituality             │    │
│    Rishikesh, Haridwar              │    │
│    Road: 6h, ₹800                   │    │
│                                          │
│ [Continue with Selection] →          │    │
│                                          │
└─────────────────────────────────────────┘
```
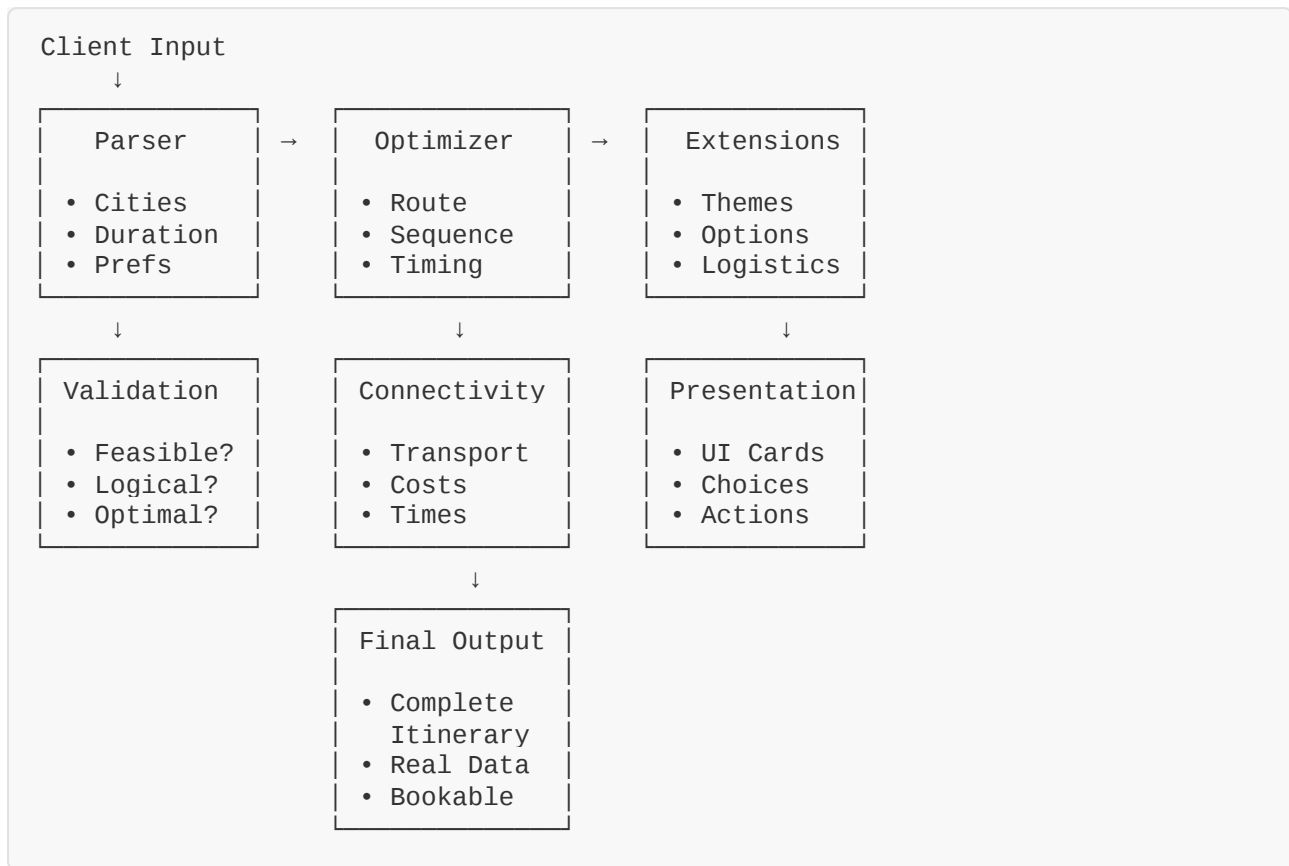
🔁 SYSTEM WORKFLOW

Backend Processing

```
1. INPUT RECEIVED
   ├── Parse destinations
   ├── Extract duration
   ├── Identify preferences
   └── Validate feasibility

2. ROUTE OPTIMIZATION
   ├── Geographical analysis
   ├── Transportation options
   ├── Time allocation
   └── Cost estimation

3. EXTENSION GENERATION
   ├── Regional mapping
   ├── Theme categorization
   ├── Connectivity check
   └── Duration calculation

4. PRESENTATION LAYER
   ├── Format options
   ├── Add visual elements
   ├── Include pricing
   └── Present choices
```

Frontend Interaction

```
1. USER INPUT
   └── Natural language processing

2. LOADING STATE
   └── "Analyzing your preferences..."

3. RESULTS DISPLAY
   ├── Optimized route
   ├── Duration breakdown
   └── Extension options

4. CHOICE SELECTION
   ├── Radio button themes
   ├── Detailed information
   └── Confirmation flow

5. FINAL ITINERARY
   └── Complete 12-day plan
```

📊 DATA FLOW DIAGRAM

```
Client Input
     ↓
┌───────────────┐     ┌───────────────┐     ┌───────────────┐
│    Parser     │  →  │   Optimizer   │  →  │  Extensions   │
│               │     │               │     │               │
│ • Cities      │     │ • Route       │     │ • Themes      │
│ • Duration    │     │ • Sequence    │     │ • Options     │
│ • Prefs       │     │ • Timing      │     │ • Logistics   │
└───────────────┘     └───────────────┘     └───────────────┘

      ↓                     ↓                     ↓
┌───────────────┐     ┌───────────────┐     ┌───────────────┐
│  Validation   │     │ Connectivity  │     │ Presentation  │
│               │     │               │     │               │
│ • Feasible?   │     │ • Transport   │     │ • UI Cards    │
│ • Logical?    │     │ • Costs       │     │ • Choices     │
│ • Optimal?    │     │ • Times       │     │ • Actions     │
└───────────────┘     └───────────────┘     └───────────────┘

                            ↓
                      ┌───────────────┐
                      │ Final Output  │
                      │               │
                      │ • Complete    │
                      │   Itinerary   │
                      │ • Real Data   │
                      │ • Bookable    │
                      └───────────────┘
```

🎯 KEY ADVANTAGES

1. Intelligent Processing

- Handles ambiguous input gracefully

- Optimizes routes geographically

- Suggests logical extensions

2. Real Data Integration

- Google connectivity information

- Actual transport costs and times

- Current availability status

3. User Choice Control

- Multiple themed options
- Clear pricing information
- Flexible duration allocation

4. Scalable Architecture

- Easy to add new destinations
- Extensible theme categories
- Modular component design

🚀 IMPLEMENTATION STATUS

✅ **Completed Components:** - Input parsing and validation - Route optimization logic - Extension suggestion engine - Regional connectivity mapping - Basic UI presentation

🔄 **In Progress:** - Google API integration - Real-time pricing updates - Advanced UI components

📋 **Next Phase:** - Cost calculation module - Booking integration - Payment processing

---

This design ensures that minimal client input ("Goa Delhi Kolkata for 12 days") is transformed into comprehensive, actionable travel plans with real logistics and multiple options for client choice.