Spica – Mini GHSE Take-Home Assignment

Spica – Mini GHSE Take-Home Assignment

Context

Spica builds location-intelligent, context-aware plans by sequencing real-world places under geographic and behavioural constraints.

This assignment is designed to understand **how you think about systems**, constraints, and sequencing.

Please do **not** spend more than **6 hours** on this.

Problem Statement

You are given:

- A starting location
- A fixed amount of time
- A set of nearby places (provided as JSON)
- User preferences and constraints

Your task is to design and implement a **small, deterministic engine** that:

1. Selects **2–3 places**
2. Decides the **order** in which they should be visited
3. Explains **why** that sequence makes sense

No external APIs are required.

Input

You will be given a JSON file in the following format:

```
{
  "user": {
    "lat": 12.9716,
```

```
    "lng": 77.5946,
    "time_available_minutes": 180,
    "preferences": ["coffee", "walk", "quiet"],
    "avoid": ["crowded"],
    "start_time": "16:30"
  },
  "places": [
    {
      "id": "p1",
      "name": "Cafe A",
      "type": "cafe",
      "lat": 12.9721,
      "lng": 77.5950,
      "avg_duration_minutes": 45,
      "crowd_level": "medium",
      "open_from": "08:00",
      "open_to": "20:00"
    },
    {
      "id": "p2",
      "name": "Park B",
      "type": "park",
      "lat": 12.9730,
      "lng": 77.5932,
      "avg_duration_minutes": 40,
      "crowd_level": "low",
      "open_from": "05:00",
      "open_to": "19:00"
    },
    {
      "id": "p3",
      "name": "Bar C",
      "type": "bar",
      "lat": 12.9698,
      "lng": 77.5961,
      "avg_duration_minutes": 60,
      "crowd_level": "high",
      "open_from": "18:00",
      "open_to": "23:00"
    },
```

```
    {
      "id": "p4",
      "name": "Bookstore D",
      "type": "bookstore",
      "lat": 12.9705,
      "lng": 77.5928,
      "avg_duration_minutes": 30,
      "crowd_level": "low",
      "open_from": "10:00",
      "open_to": "21:00"
    }
  ]
}
```

Distance can be calculated using a simple Haversine formula or a reasonable approximation.

Output

Your program should output:

- An **ordered list of place IDs**
- **Total time spent**
- A brief **reasoning for each step**

Example:

```
{
  "sequence": ["p2", "p1", "p4"],
  "total_time_minutes": 150,
  "explanation": {
    "p2": "Park chosen first as it’s quiet and nearby before sunse
t",
    "p1": "Cafe next to rest after walk; crowd level acceptable",
    "p4": "Bookstore fits remaining time and preference for quiet"
  }
}
```

Console output or a JSON file is sufficient.

Required Explanation (README or comments)

Please answer the following:

1. What constraints mattered most in your decision-making and why?
2. What constraints did you intentionally ignore or simplify?
3. What would break if the number of places doubled?
4. How would your approach change for a friend group instead of a single user?

Optional extension (no code required):

If this sequencing logic were to be integrated into a mobile app (e.g., React Native), briefly explain:

- where this logic should live (client vs server),
- how the app would interact with it (API shape / data flow), and
- what constraints you would consider (latency, offline usage, errors, state consistency).

Explicit Limitation

State **one limitation** of your solution.

Example:

> “This approach does not adapt well to real-time changes in crowd levels.”

Guidelines

- Any language is allowed (Node.js / Python preferred)
- No UI required
- External APIs are **not** required
- AI tools may be used for syntax or boilerplate, but **decision logic must be your own**

Evaluation Criteria

We will evaluate based on:

- Clarity of reasoning
- How constraints are modeled

- Sequencing logic
- Tradeoff awareness
- Ability to articulate limitations
- (Optional) Thoughtfulness around app integration

We value clarity of reasoning over completeness.