



Project Title: “Smart Travel Planner & Cost Optimizer ✈️🏠”



Project Overview:

You are building a **Smart Travel Planner** that helps users plan **multi-city trips** in the most **cost-efficient way**.

The system will suggest the **optimal sequence of cities to visit** to minimize total travel cost & maximize enjoyment points under budget and time constraints.

This project simulates:

- Flight/train costs between cities (2D DP - Matrix Based).
- Hotel stays in each city (1D DP - Daily Cost).
- Enjoyment points per city (Maximization Objective).



Problem Modeling:



Problem Description:

- A traveler has a fixed **budget** 💰 and **maximum trip days** 📅.
- There are **N cities** to visit.
- Each city has:
 - 🏠 **Hotel stay cost per night.**
 - 🎡 **Enjoyment points per day.**
- Travel between cities costs **different amounts** (2D Cost Matrix).
- The traveler starts from a fixed **starting city** and must end at a specific **destination city**.



Objectives:

1. Minimize Total Cost (Hotels + Travel).
2. Maximize Total Enjoyment Points (Within Budget & Time Limit).
3. Select Best Possible Route & Days in Cities.

Dynamic Programming Breakdown:

Module	DP Type Used	Description
Hotel Stay Cost Optimization	1D DP	Optimize daily costs & enjoyment points within budget per city.
Route Planning Between Cities	2D DP	Solve "Traveling Salesman"-like route planning with Min Cost Path DP (Matrix).
Combined DP Optimization	Mix (1D + 2D)	Combine hotel costs & route costs to minimize total expenses while maximizing points.

Key Steps (High-Level Algorithm):

1. **Step 1:**
For each city, use **1D DP** to compute:
 - Best number of days to stay → Max enjoyment points within hotel budget limit.
2. **Step 2:**
Use **2D DP** (Graph-like DP) on **cost matrix** to:
 - Find the cheapest travel route visiting selected cities (Min Cost Path between cities).
3. **Step 3:**
Merge results to compute:

- Total trip cost (Hotel Stay + Travel).
- Total enjoyment points.



Sample Scenario Example:

City	Hotel Cost/Night	Enjoyment Points/Day
------	------------------	----------------------

A	₹2000	5
---	-------	---

B	₹1500	4
---	-------	---

C	₹3000	8
---	-------	---



Travel Cost Matrix (₹):

From/To	A	B	C
A	0	1000	2000
B	1000	0	1200
C	2000	1200	0



Constraints:

- Budget: ₹15,000
- Max Trip Days: 7
- Start City: A

- End City: C

Goal: Maximize enjoyment points within the budget.



Expected Output Example:



Best Route: A → B → C



Days in Cities: A(2), B(2), C(3)



Total Trip Cost: ₹14,800



Total Enjoyment Points: 39



Advanced Features (Optional for Extra Credit):

- ☒ Add option to skip cities.
- ☒ Allow "round trip" (Start & End same city).
- ☒ Support "multiple travelers" with shared hotels.
- ☒ Visualize route using ASCII maps or graphs.



Why It's Challenging & Realistic:

- Combines both **1D & 2D DP** in a real-world problem.
- Closely related to **Knapsack, Traveling Salesman Problem, & Path Finding**.
- Highly extensible for advanced students.
- Mimics complex **route optimization systems** used in travel apps.



Learning Outcomes:

- Solve **multi-objective optimization** problems.
- Apply **1D & 2D DP** together.

- Practice matrix DP (Graph-like problems).
- Improve system design & modularization.
- Prepare for advanced algorithmic interviews.



Project Deliverables:

1. ☒ Clean modular Python code (functions/classes).
2. ☒ Explanation of all DP states, transitions & base cases.
3. ☒ Readable documentation/report (PDF or Markdown).
4. ☒ Sample Input/Output Data Files.
5. ☒ Optional CLI/GUI Interface (for bonus).