**Advanced Python programming**

**Lab 3**

**NAME:S.JEWEL REDDY**

**REG NO:22MID0161**

**10 Shared Features (Similarities) GoLobe (fullstack‑nextjs‑golobe‑travel‑agency)**:

| **Feature Area** | **My Code** | **Open‑Source OTAs** |
| --- | --- | --- |
| **1.Destination Lists** | Uses append(), insert(), sort(), pop() | Maintains destination catalogs (arrays/DB queries) [RPubs+2GitHub+2GitHub+2](https://github.com/scriptandtools/Online-Travling-System-Php?utm_source=chatgpt.com) |
| 2. **User Preference Map** | Maps interests → destinations (interest\_map) | Stores user choices/preferences during booking process [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com)[GitHub](https://github.com/eli64s/python_recommendation_engine?utm_source=chatgpt.com) |
| 3. **Collaborative Filtering** | Simple logic based on shared interests | Applies CF on travel history to recommend destinations [GitHub](https://github.com/eli64s/python_recommendation_engine?utm_source=chatgpt.com) |
| 4. **Interactive User Input** | Uses input() prompts | OTAs use form-based user input for travel needs [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com) |
| 5. **Itinerary Generation** | Creates multi-destination sample itinerary | OTA UIs generate booking itineraries/flights/hotels list [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com) |
| 6. **Modular Code Design** | Functions like recommend\_by\_interest() | Microservices (catalog, booking, auth) in OTAs [GitHub](https://github.com/otasoft?utm_source=chatgpt.com) |
| 7. **Sorting & Ranking** | Sorts destinations.sort() alphabetically | Rankings based on price/popularity in OTA catalog [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com) |
| 8. **Duplicate Removal** | set() for uniqueness | DB-level deduplication in offer catalogs [GitHub](https://github.com/otasoft?utm_source=chatgpt.com) |
| 9. **Filter-Based Selection** | Removes irrelevant options via pop() | Offers filtering (budget, stars, reviews) in OTA search [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com) |
| 10. **Extendability** | Easily add new filters/interests | OTAs offer plugin-friendly microservices |

**10 Key Differences**

**In Code Design & Architecture**

1. **Tech Stack**
   * *My code*: Pure Python script (CLI-style)
   * *GoLobe OTA*: React/NextJS frontend, MongoDB backend [GitHub+1GitHub+1](https://github.com/topics/travel-agency?utm_source=chatgpt.com)
2. **Data Persistence**
   * *My code*: In-memory lists
   * *GoLobe*: Persistent storage (MongoDB), user accounts [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com)
3. **Real-Time Data Integration**
   * *My Code*: Hardcoded destination and interest map
   * *GoLobe*: Dynamic flight/hotel search via OTA-styled APIs [GitHub+5GitHub+5GitHub+5](https://github.com/topics/travel-agency?utm_source=chatgpt.com)[GitHub+1Wikipedia+1](https://github.com/topics/travel-agency-website?l=javascript&o=asc&s=stars&utm_source=chatgpt.com)
4. **Microservice Architecture**
   * *My Code*: Monolithic script
   * *Otasoft eco-system*: Modular services for catalog, booking, auth [GitHub](https://github.com/otasoft?utm_source=chatgpt.com)
5. **Scalable CF**
   * *My Code*: List-based matching
   * *python\_recommendation\_engine*: Matrix-based CF using user history [Kaggle](https://www.kaggle.com/code/amalshajiprof/recommender-system-for-travel-packages-beginner?utm_source=chatgpt.com)[Wikipedia+2GitHub+2Stack Overflow+2](https://github.com/eli64s/python_recommendation_engine?utm_source=chatgpt.com)

**In Collaborative Features**

1. **User Similarity**
   * *My Code*: Shared interests list
   * *python\_recommendation\_engine*: Actual flight origins/destinations history [GitHub](https://github.com/eli64s/python_recommendation_engine?utm_source=chatgpt.com)
2. **Implicit vs Explicit Feedback**
   * *My Code*: Explicit yes/no on interests
   * *python\_recommendation\_engine*: Uses implicit behavior (travel history) [Wikipedia+1Real Python+1](https://en.wikipedia.org/wiki/Collaborative_filtering?utm_source=chatgpt.com)
3. **Cold-Start Handling**
   * *My Code*: No specific mechanism
   * *Reference CF systems*: Use survey or initial ratings [Wikipedia](https://en.wikipedia.org/wiki/MovieLens?utm_source=chatgpt.com)

**In Business Intelligence**

1. **Pricing & Offers**
   * *My Code*: No pricing logic
   * *OTAs*: Real-time pricing, bundles, dynamic offers [GitHub](https://github.com/topics/travel-agency?utm_source=chatgpt.com)[GitHub](https://github.com/otasoft?utm_source=chatgpt.com)
2. **Analytics & ML**

* *My Code*: Simple mapping
* *python\_recommendation\_engine*: Can log usage statistics vs models
* *Otasoft-ml module*: Supports ML for better suggestions [GitHub+1Umair Iftikhar+1](https://github.com/otasoft?utm_source=chatgpt.com)

**Links to the Reference OTAs**

* **GoLobe (fullstack‑nextjs‑golobe‑travel‑agency)**: GitHub project with search & booking logic [GitHub+2GitHub+2GitHub+2](https://github.com/topics/travel-agency?utm_source=chatgpt.com)
* **Python Recommendation Engine (eli64s)**: CF-based notebook using travel history [GitHub](https://github.com/eli64s/python_recommendation_engine?utm_source=chatgpt.com)

**Collaborative Highlights**

* **My Coder s** uses basic CF: interest\_map + list ops → recommendation
* **Eli64s notebook** uses **real travel history** and CF algorithms (user‑based, implicit)
* **OTAs** feed recommendation modules via centralized catalog, user data, A/B testing and ML pipelines

**Summary**

* My Code share many **list operations and filtering logic** with OTAs.
* Major differences lie in **architecture, real-time integration, scalability**, and **business analytics**.
* Collaborative filtering in My Coder code is introductory; open-source solutions are production-grade, handling **cold-start**, **implicit behavior**, **analytics tracking**, and **microservices**.