

## <새로 추가된 기능과 구현 방법>

### 1. 세부적인 가격 분석

- **설명:** 기존 평균 가격만 출력하는 대신, 항공권의 최저 가격과 최고 가격도 함께 출력하여 예산의 유동성을 고려할 수 있도록 개선함.
- **구현 방법:**
  - FlightData 클래스에 getLowestPrice 및 getHighestPrice 메서드를 추가.
  - getLowestPrice 는 std::min\_element 를 사용하여 최저값 반환.
  - getHighestPrice 는 std::max\_element 를 사용하여 최고값 반환.
  - PricePredictor 의 suggestDestinations 메서드에서 평균, 최저, 최고 가격을 함께 출력하도록 수정.

### 2. 상세 추천 메시지

- **설명:** 추천 여행지 출력 시 "왜 이 여행지가 추천되는지"에 대한 설명을 추가함
- **구현 방법:**
  - PricePredictor::suggestDestinations 메서드에서 조건을 확인 후 해당 여행지가 예산 범위 내에 포함되는 이유를 명확히 출력.

### 3. 데이터 관리 편의성

- **설명:** 사용자가 프로그램 실행 중에도 새로운 항공권 데이터를 추가하거나 특정 데이터를 삭제할 수 있도록 기능을 추가함
- **구현 방법:**
  - PricePredictor 클래스에 addFlightData 메서드 외에 removeFlightData 메서드 추가.
  - removeFlightData 는 std::remove\_if 를 사용하여 특정 국가 데이터를 삭제.

### 4. 최저가 여행지 추천 기능

- **설명:** 예산 내에서 가장 저렴한 항공권이 있는 여행지를 따로 추천함
- **구현 방법:**
  - PricePredictor 클래스에 suggestLowestPriceDestination 메서드 추가.
  - getLowestPrice 메서드로 항공권의 최저 가격을 확인하여, 예산 내 가장 저렴한 여행지 선택.
  - std::numeric\_limits<int>::max()를 사용해 초기값을 설정하고, 반복문으로 최저가 항공권을 가진 데이터를 탐색.

## 5. 추천 월과 예산 필터 통합

- **설명:** 사용자가 특정 월에 여행을 원할 경우, 해당 월에 추천되는 여행지를 필터링하여 출력.
- **구현 방법:**
  - PricePredictor 클래스에 suggestByMonth 메서드 추가.
  - std::find를 사용하여 FlightData의 bestMonths에서 사용자가 입력한 월이 포함되어 있는지 확인.
  - 조건에 맞는 여행지만 출력.

## 6. 데이터 정렬 기능

- **설명:** 추천 리스트를 항공권 평균 가격 기준으로 정렬하여 출력.
- **구현 방법:**
  - std::sort를 사용하여 PricePredictor의 flightDataList를 평균 가격 기준으로 오름차순 정렬.
  - 정렬은 suggestDestinations 메서드 내부에서 수행.

## <테스트 결과>

```
Enter your travel budget ($): 500
Enter your trip duration (days): 3
Enter your preferred travel month (optional, press enter to skip): 8

Recommended Destinations within your budget:
Country: Korea
Average Price: $200
Lowest Price: $180 | Highest Price: $220
Best Months to Visit: April May September
-----

Cheapest Destination within your budget:
Country: Korea
Lowest Price: $180
Best Months to Visit: April May September

Destinations Recommended for 8:
No destinations available for the month of 8.
```

## <피드백>

사용자 입력에 대한 오류 처리 부족

- ➔ 사용자가 잘못된 값을 입력(예: 음수 예산, 너무 짧은 기간 등)해도 프로그램이 제대로 처리하지 않고 진행됨