

C++프로그래밍 및 실습

데이트 코스 추천 프로그램

진척 보고서 #1

제출일자:2024.11.06.

제출자명:손명현

제출자학번:234124

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

현대인은 일상에서 소중한 사람들과의 만남을 준비하며 장소와 활동을 찾는 데 고민과 시간이 많이 소모됨. 친구들과의 모임에서 "어디 가지?", "뭐 먹지?" 같은 질문으로 시간을 낭비하거나 만족스럽지 못한 결과를 겪는 경우가 많음. 또한, 가족 단위의 모임 시 인스타 감성에 치우친 장소나 지나치게 시끄러운 분위기 탓에 가족모임에 어울리지 않는 곳을 방문하여 불만족스러운 경험을 하게 되는 경우가 있음. 이러한 문제 해결을 위해, 다양한 모임 형태(친구, 가족, 연인 등)에 맞춤형 데이트 코스를 추천하는 프로그램이 필요함. 이 프로그램은 사용자가 시간을 절약하고 모임의 목적과 분위기에 맞는 최적의 코스를 추천받아 만족스러운 경험을 할 수 있도록 돕는 것을 목표로 함.

2) 프로젝트 목표

본 프로젝트의 목표는 사용자가 모임의 종류와 목적에 맞는 맞춤형 데이트 코스를 제공하는 것임. 사용자가 모임의 종류(친구, 가족, 연인 등)와 원하는 지역을 선택하면, 해당 모임에 어울리는 장소와 코스를 제안하는 프로그램을 개발할 예정임. 이 프로그램은 모임 특성과 장소의 분위기, 접근성 등을 종합적으로 고려하여 다양한 데이트 코스를 제공함으로써 사용자의 만족도를 높이하고자 함.

3) 차별점

현재 유사한 어플리케이션은 주로 20~30대를 타겟으로 하여 인스타 감성을 지향하고 있어, 가족모임처럼 다른 분위기를 선호하는 사용자들에게는 불편함이 있음. 본 프로그램은 이러한 문제를 해결하기 위해 다음과 같은 차별점을 가짐.

모임 특성 기반 추천: 친구, 가족, 연인 등 모임의 목적과 특성에 맞춰 장소와 코스를 추천함. 예를 들어, 가족 모임에는 조용하고 여유로운 분위기의 장소를, 친구 모임에는 활동적인 장소를 제안하여 각 모임 성격에 맞는 데이트 코스를 제공할 예정임.

장소 분위기와 걷기 난이도 고려: 모임 특성에 따라 장소의 분위기(조용함, 활기참 등)와 걷기 편의성 등 다양한 요소를 고려하여 코스를 구성함. 예를 들어, 연인 코스에는 산책하기 좋은 코스를 추천하고, 가족 단위의 코스에는 장시간 걷지 않아도 다양한 즐길 거리가 있는 장소를 제안함.

사용자 경험 중심의 맞춤형 코스 제공: 단순히 유명한 장소를 나열하는 것이 아닌, 사용자의 취향과 모임 특성에 따라 코스를 구성해 더욱 풍부한 경험을 제공한다.

2. 기능 계획

1) 기능 1 맞춤형 데이트 코스 추천

- 설명 사용자가 모임의 종류와 위치를 선택하면, 이에 맞춰 최적화된 데이트 코스를 추천하는 기능. 친구 모임, 연인 데이트, 가족 모임 등 다양한 유형에 따라 장소와 활동을 맞춤

(1) 세부 기능 1 모임 유형에 따른 필터링

사용자가 선택한 모임 유형에 따라 분위기, 소음 정도, 걷기 편의성 등의 기준을 설정하여 장소를 필터링. 예를 들어, 가족 모임을 위한 코스는 조용하고 편안한 장소를, 친구 모임은 활기찬 장소를 추천함.

3. 진척사항

1) 기능 구현

(1) 구현한 기능 이름

- 입출력 : 사용자가 모임의 종류(가족, 애인, 친구) 중 하나를 선택하도록 입력을 받고, 선택에 따라 맞춤형 메시지를 출력하는 기능.

- 설명 : 프로그램은 콘솔을 통해 사용자와 상호작용하며, 사용자가 입력한 숫자(1, 2, 3)에 따라 적절한 모임 종류를 인식하여 메시지를 제공함. 올바르지 않은 입력에는 안내 메시지를 출력함.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등) :

입출력: cin, cout를 사용해 사용자와 상호작용.

조건문: switch문으로 사용자의 입력값에 따라 다른 동작 수행.

기초 데이터 타입: int를 사용해 숫자 입력 처리.

- 코드 스크린샷

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int choice;
6
7      cout << "모임 종류 선택하세요:\n";
8      cout << "1. 가족\n";
9      cout << "2. 애인\n";
10     cout << "3. 친구\n";
11     cout << "선택 (숫자 입력): ";
12     cin >> choice;
13
14     switch (choice) {
15         case 1:
16             cout << "가족 모임을 위한 추천 코스를 준비 중입니다.\n";
17             // 가족 모임에 적합한 코스 추천 기능 추가 가능
18             break;
19         case 2:
20             cout << "애인을 위한 추천 코스를 준비 중입니다.\n";
21             // 애인 모임에 적합한 코스 추천 기능 추가 가능
22             break;
23         case 3:
24             cout << "친구들과 함께 할 추천 코스를 준비 중입니다.\n";
25             // 친구 모임에 적합한 코스 추천 기능 추가 가능
26             break;
27         default:
28             cout << "올바른 숫자를 입력해 주세요.\n";
29             break;
30     }
31
32     return 0;
33 }
```

2) 테스트 결과

(1) 테스트한 기능 이름

- 설명

입력값	출력 결과	기대 결과	테스트 결과
1	가족 모임을 위한 추천 코스를 준비 중입니다.	가족 관련 메시지 출력	성공
2	애인을 위한 추천 코스를 준비 중입니다.	애인 관련 메시지 출력	성공
3	친구들과 함께 할 추천 코스를 준비 중입니다.	친구 관련 메시지 출력	성공
0	올바른 숫자를 입력해 주세요.	잘못된 입력 메시지 출력	성공
a (문자)	올바른 숫자를 입력해 주세요.	잘못된 입력 메시지 출력	성공

- 테스트 결과 스크린샷

```
모임 종류를 선택하세요:
1. 가족
2. 애인
3. 친구
선택 (숫자 입력): 1
가족 모임을 위한 추천 코스를 준비 중입니다.
```

```
모임 종류를 선택하세요:
1. 가족
2. 애인
3. 친구
선택 (숫자 입력): 2
애인을 위한 추천 코스를 준비 중입니다.
```

```
모임 종류를 선택하세요:
1. 가족
2. 애인
3. 친구
선택 (숫자 입력): 3
친구들과 함께 할 추천 코스를 준비 중입니다.
```

```
모임 종류를 선택하세요:
1. 가족
2. 애인
3. 친구
선택 (숫자 입력): 0
올바른 숫자를 입력해 주세요.
```

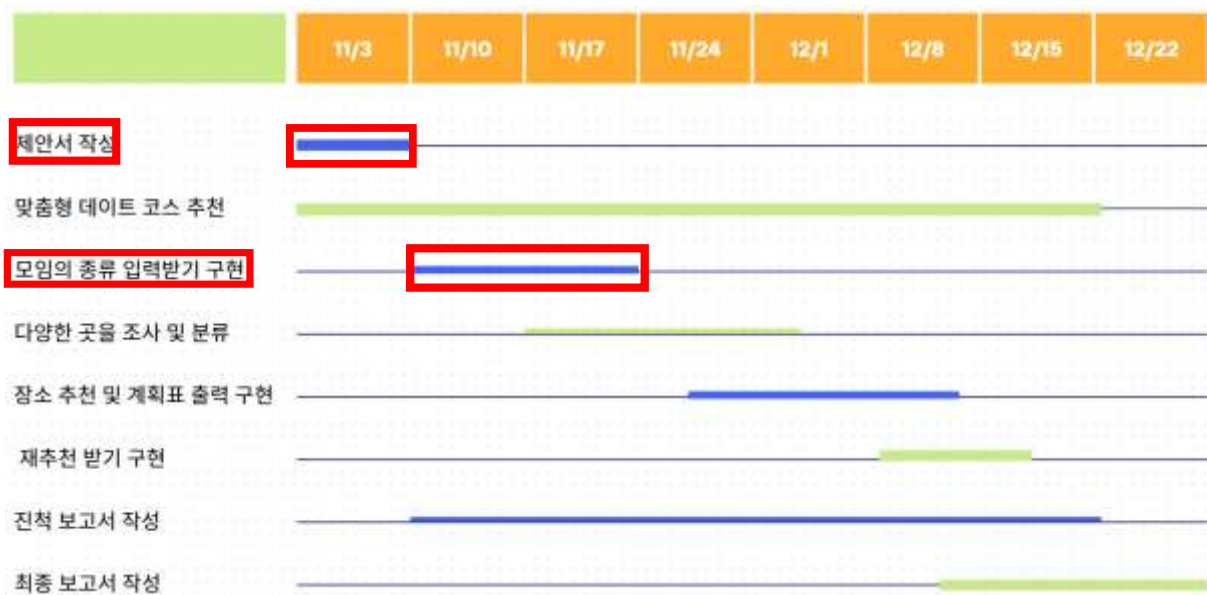
모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): a

올바른 숫자를 입력해 주세요.

3. 프로젝트 일정 (참고: 간트차트)



완료시 ☐ 로 표시