도서관 좌석 추천 시스템

group S

LIBRARY SMART SEAT FINDER

김태은 2017*****

김두용 202135307

박세렴 202334469

홍영철 202139857

con

01 — TOPIC

02 — IMPLEMENTATION

03 — HOW TO USE

04 — LIMITATIONS & DEVELOPMENTS

tents

group S

01 TOPIC

01 TOPIC

SEAT RECOMMANDATION

Traditional library reservation systems

+

select options for seats

01 Options for Location

02 Select Option

03 Recommended Seat Highlight Visualization

LOCATION OPTION

- corner : Check the column of the block
- adjacent: Empty on bothsides, additional inspection when corner seats
- window : Check the column close to the window

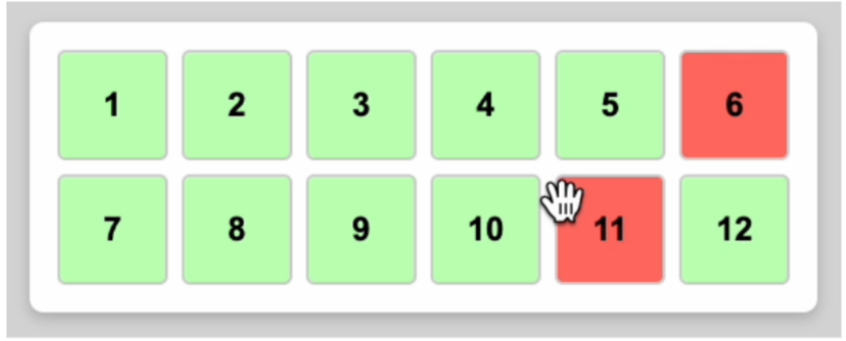
SELECT OPTION

- AND : Satisfy all selected conditions
- OR : Satisfy at least one selected condition

Types of Options

02
IMPLEMENTATION

UI & DESIGN

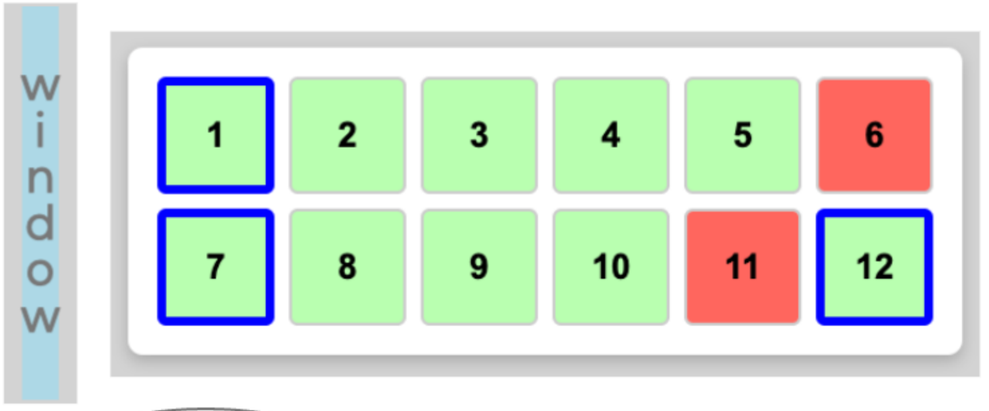


Seating arrangement

Implementation to dynamically create seat blocks and rows

Mouse

Add mouse hover animation to seat



Window

Add window box to the left of the screen

Animation

Anime that emphasizes seat recommendations sequentially

Recommendation Algorithm

1 · 2 · 3 · 4

Weighted by seat
- corner
- adjacent
- window

Check the location conditions for each seat

Check additional conditions
- AND
- OR

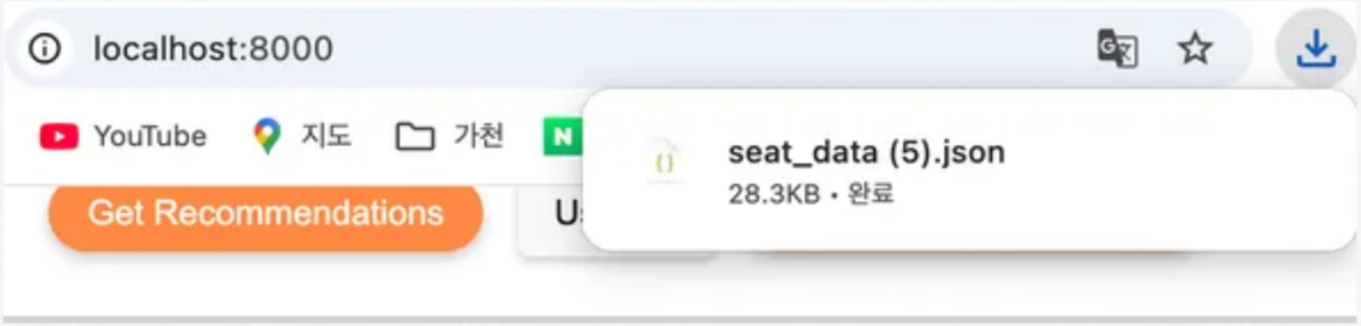
Calculate each weight score
- Score above 0 and recommended after checking AND/OR conditions

Seat Recommendation

Reserve and Save

Recommendation Algorithm

Reserve and Save



Local file modification is not possible due to html limitations,
so new seat file is downloaded

Main Code explanation

```
const isCorner = seat.col === seat.blockStartCol || seat.col === seat.blockEndCol;
if (isCorner) {
  score += weights.corner;
} else if (useAndCondition) {
  allConditionsMet = false;
}
```

1. Verify corner with iscorner variable
2. If you fall into the corner, add as many points as weights.corner
3. For AND conditions, set allConditionsMet to false if the condition is not met

```
const leftSeat = seats.find(s => s.row === seat.row && s.col === seat.col - 1 && s.block === seat.block);
const rightSeat = seats.find(s => s.row === seat.row && s.col === seat.col + 1 && s.block ===
seat.block);
const isLeftAvailable = !leftSeat || leftSeat.status === "available";
const isRightAvailable = !rightSeat || rightSeat.status === "available";

if (isLeftAvailable || isRightAvailable) {
  score += weights.adjacent;
} else if (useAndCondition) {
  allConditionsMet = false;
}
```

1. Find left and right Seat to check the condition of the seat
2. If the left (isLeftAvailable) and right (isRightAvailable) seats are empty, add a score as much as weights.adjacent
3. For AND conditions, set allConditionsMet to false if the condition is not met

Main Code explanation

```
if (seat.col === 1) {
  score += weights.window;
} else if (useAndCondition) {
  allConditionsMet = false;
}
```

1. Verify that the seat is located in the first row (seat.col === 1) (close to the window)
2. Add scores as much as weights.window if the conditions are met
3. For AND conditions, set allConditionsMet to false if the condition is not met

```
return score;
```

1. Return the weighted score, recommending the seat in the order in which the score is high

03
HOW TO USE

GITHUB

https://github.com/HongYoungChul/TEAM/tree/main/Seat_Recommend

Connection

localhost:8000

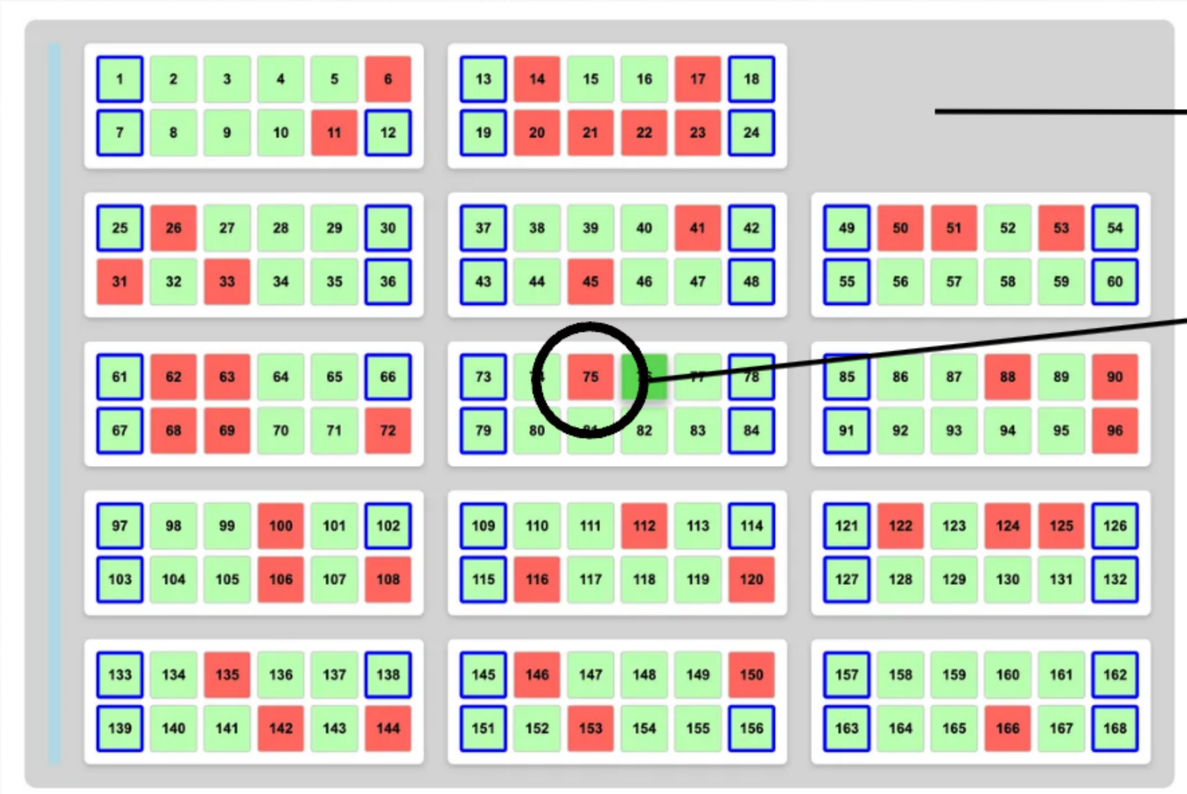
HOW TO USE

구석자리(끝자리) | 옆자리가 비어있는 자리 | 창가자리

Select location options

Get Recommendations | Use OR | Reset Recommendations

Select additional options



Get recommendations

Select a seat

Reserve

Reserve

group S

04 LIMITATIONS & DEVELOPMENTS

LIMITATIONS & DEVELOPMENTS

1 Server-use database

Now locally, update new files directly after each turn

-> Update the real-time information of the seat by connecting with the current seat on the database server

2 Diversifying Algorithms

Algorithm to set weights for each seat in the initial setting

-> Add conditions such as 'specific seat criteria' using distance-based Dijkstra algorithms to provide a more accurate basis for recommendations

3 Adding UX Features

-Change the recommended color of the seat according to the recommended priority

- Learn your own seat history through the login function

4 Positioning the seat on the administrator's side

- Features for easy modification of added or missing seats

THANK YOU