Venue Availability System

Streamlining Venue Reservations for Ateneo CFMO

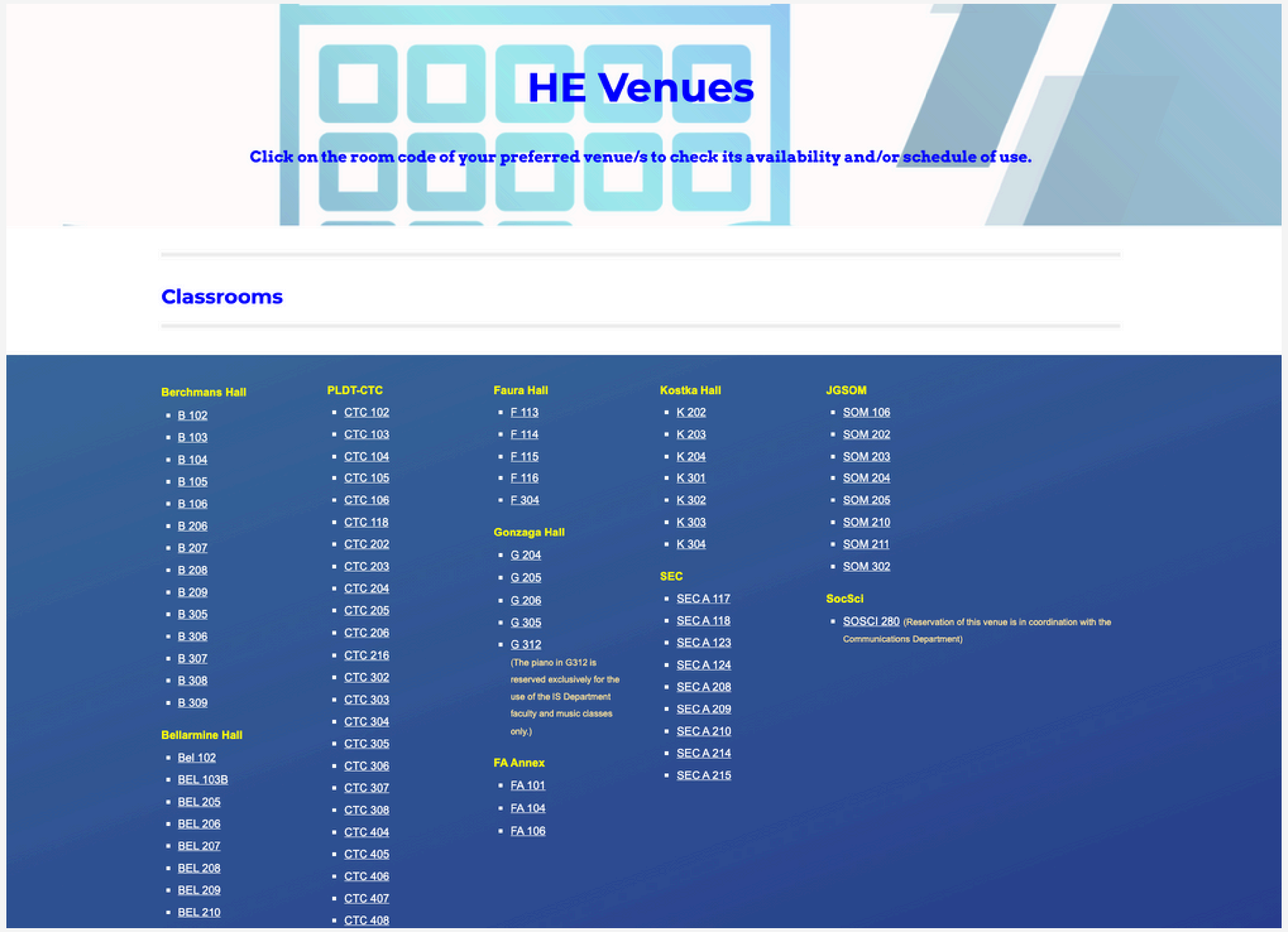
Jana Almira Boco, Nathan Luna, Juliana Valdez

MSYS 30 Final Project

Problem Statement & Context

Current Issues:

- Delays in booking due to inefficiencies.
- Overbooking caused by lack of transparency.
- Manual Looking at different venues



HE Venues

Click on the room code of your preferred venue/s to check its availability and/or schedule of use.

Classrooms

Berchmans Hall

- B 102
- B 103
- B 104
- B 105
- B 106
- B 206
- B 207
- B 208
- B 209
- B 305
- B 306
- B 307
- B 308
- B 309

Bellarmine Hall

- Bel 102
- BEL 103B
- BEL 205
- BEL 206
- BEL 207
- BEL 208
- BEL 209
- BEL 210

PLDT-CTC

- CTC 102
- CTC 103
- CTC 104
- CTC 105
- CTC 106
- CTC 118
- CTC 202
- CTC 203
- CTC 204
- CTC 205
- CTC 206
- CTC 216
- CTC 302
- CTC 303
- CTC 304
- CTC 305
- CTC 306
- CTC 307
- CTC 308
- CTC 404
- CTC 405
- CTC 406
- CTC 407
- CTC 408

Faura Hall

- F 113
- F 114
- F 115
- F 116
- F 304

Gonzaga Hall

- G 204
- G 205
- G 206
- G 305
- G 312

(The piano in G312 is reserved exclusively for the use of the IS Department faculty and music classes only.)

FA Annex

- FA 101
- FA 104
- FA 106

Kostka Hall

- K 202
- K 203
- K 204
- K 301
- K 302
- K 303
- K 304

SEC

- SEC A 117
- SEC A 118
- SEC A 123
- SEC A 124
- SEC A 208
- SEC A 209
- SEC A 210
- SEC A 214
- SEC A 215

JGSOM

- SOM 106
- SOM 202
- SOM 203
- SOM 204
- SOM 205
- SOM 210
- SOM 211
- SOM 302

SocSci

- SOSCI 280 (Reservation of this venue is in coordination with the Communications Department)

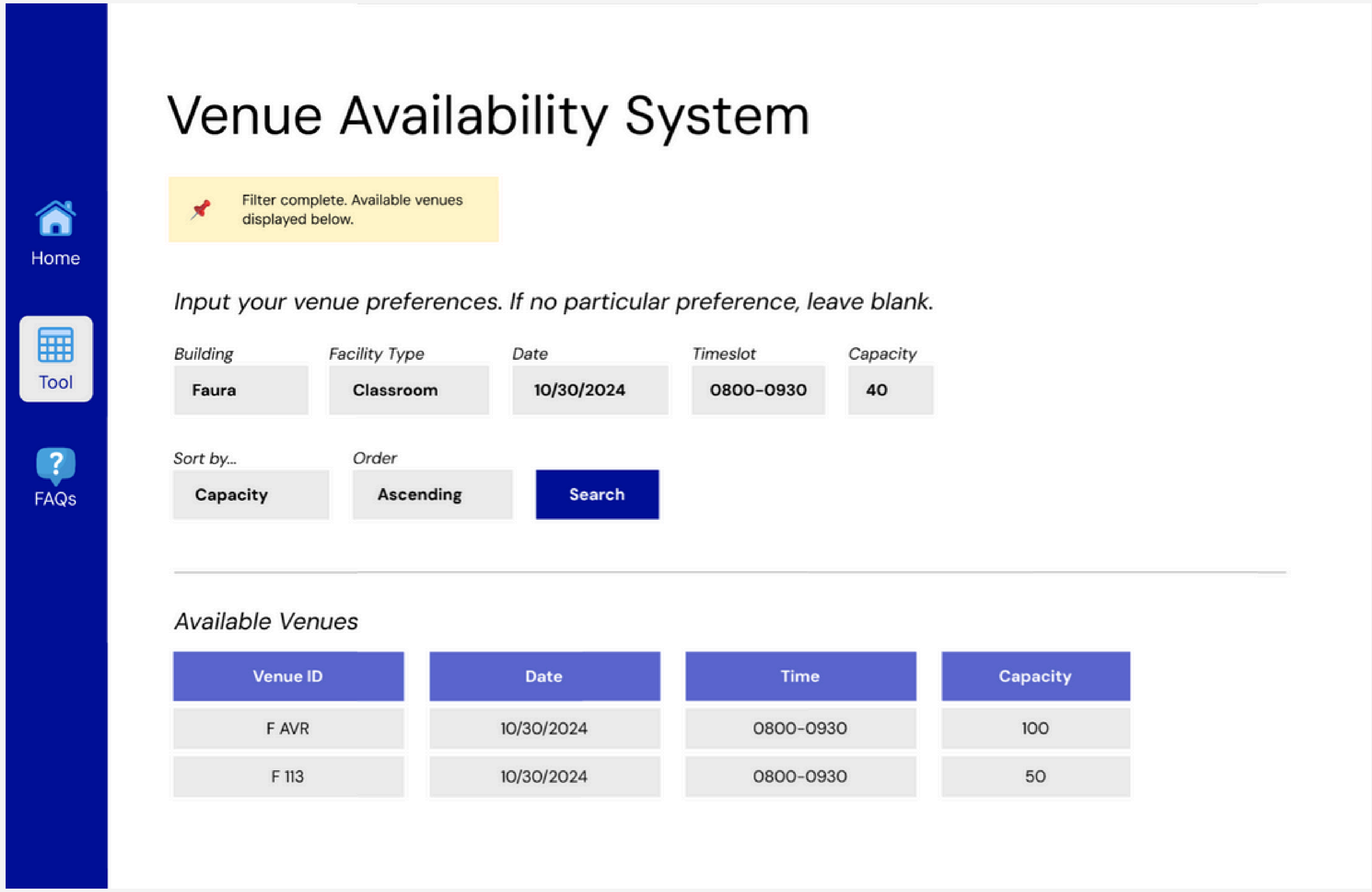
Goal: : Create a consolidated venue availability system for students.

Back to Agenda Page

MSYS 30 Final Project

Objective

- Centralized monitoring system for venue availability.
- Integration of search and sort functions for user preferences.
- Recommendations for suitable venues based on activity needs.



Venue Availability System

📌 Filter complete. Available venues displayed below.

Input your venue preferences. If no particular preference, leave blank.

Building	*Facility Type*	*Date*	*Timeslot*	*Capacity*
Faura	Classroom	10/30/2024	0800-0930	40

Sort by...: Capacity

Order: Ascending

Available Venues

Venue ID	Date	Time	Capacity
F AVR	10/30/2024	0800-0930	100
F 113	10/30/2024	0800-0930	50

Back to Agenda Page

Scope and Features

Back to Agenda Page

Scope

- Covers availability, not the booking process.
- Venues focused on Faura Hall
- Date Range
    - Jan. 15 to Feb. 14, 2025
- Limited Dropdown selection
- Dataset of 1000 records from Mockaroo

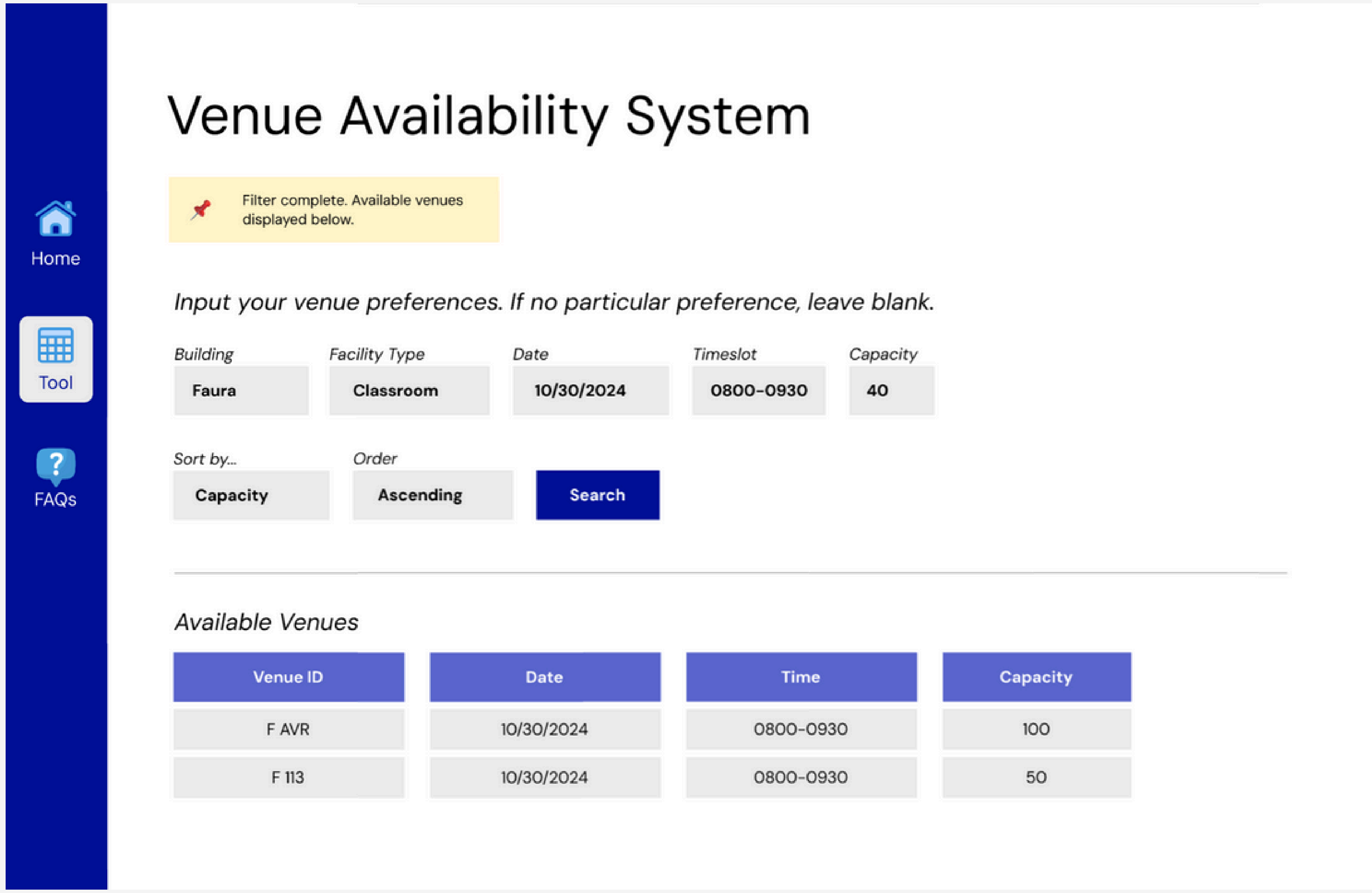
Features

- Inputs: Date, time, room capacity, sort requirements
- Outputs: Tabular listing of available rooms with detailed attributes.

MSYS 30 Final Project

Methodology

- Problem Identification and Requirements Analysis
- Data Extraction and Simulation (MOCK_DATA.csv)
- Algorithm Implementation:
  - Binary Search and MergeSort
- UI Design and Integration of Input Features
- Testing and Evaluation:
  - Functional, Performance, Edge Cases



Venue Availability System

📌 Filter complete. Available venues displayed below.

Input your venue preferences. If no particular preference, leave blank.

Building	*Facility Type*	*Date*	*Timeslot*	*Capacity*
Faura	Classroom	10/30/2024	0800-0930	40

Sort by...	*Order*
Capacity	Ascending

Available Venues

Venue ID	Date	Time	Capacity
F AVR	10/30/2024	0800-0930	100
F 113	10/30/2024	0800-0930	50

Design Decisions

Back to Agenda Page

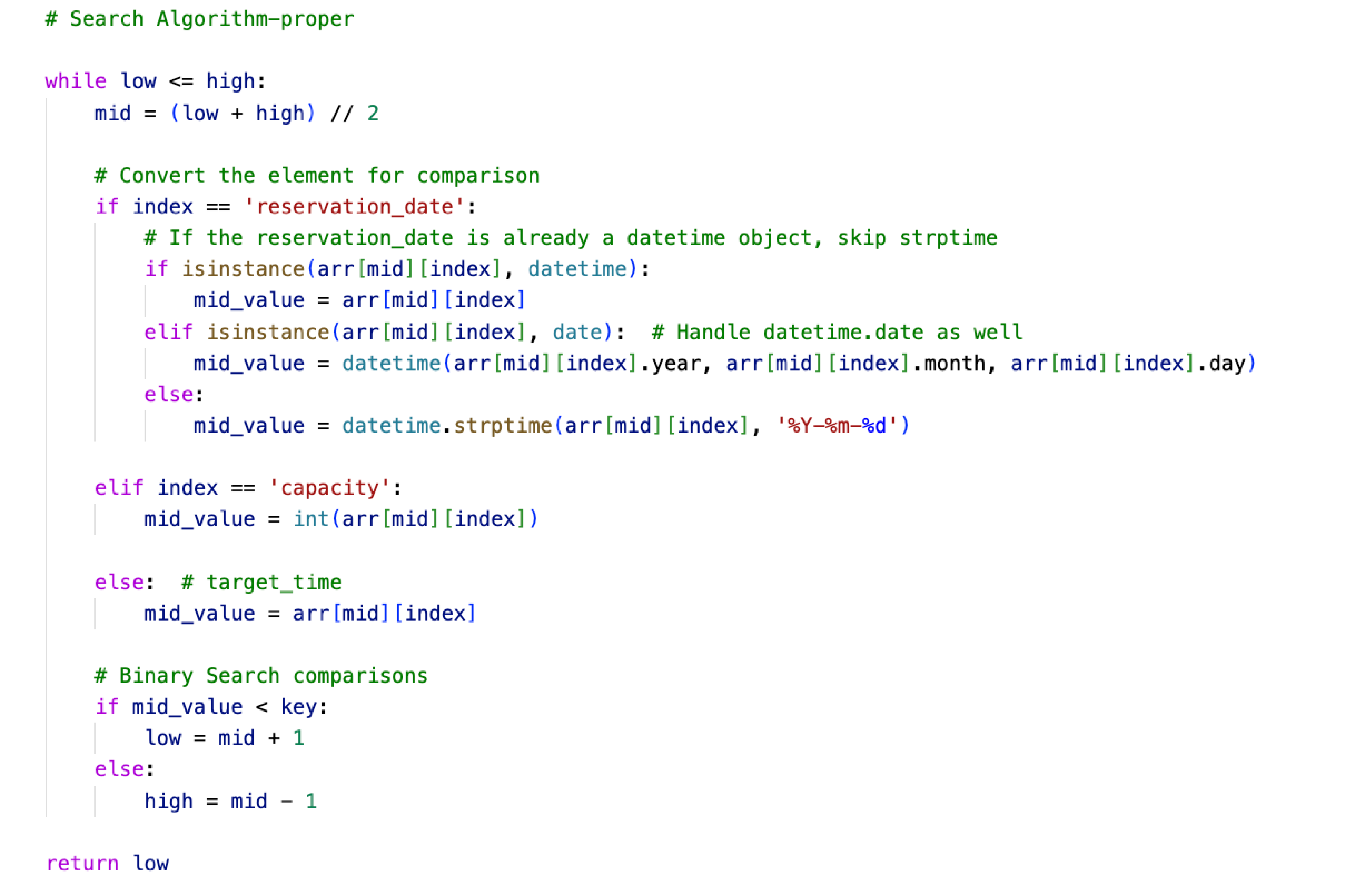
Data Structures

- **Dictionaries:**
  - Store venue details in key-value pairs for efficient access.
    - {"reservation_date": "2025-01-15", "capacity": 50}.
  - Used in: available_slots, slots, cleaned_data, and context
- **Lists**:
  - Store collections of venue data for sorting and filtering.
    - Example: time slots
  - Used in: available_slots, TIME_SLOTS, left and right in merge_sort, merged in merge, filtered_slots, forms.py

Search Algorithm

Back to Agenda Page

Binary Search



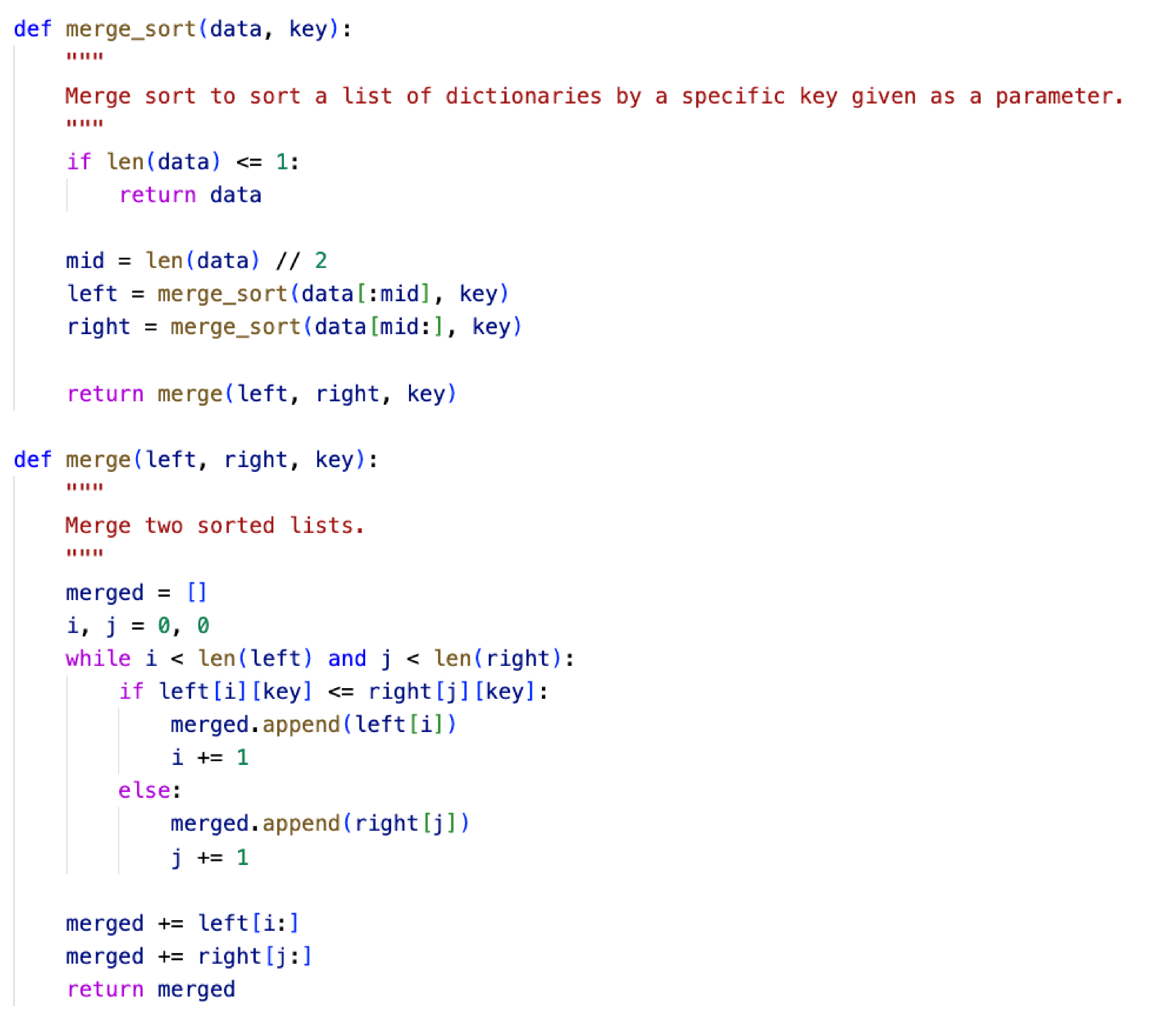
```python
# Search Algorithm-proper

while low <= high:
    mid = (low + high) // 2

    # Convert the element for comparison
    if index == 'reservation_date':
        # If the reservation_date is already a datetime object, skip strptime
        if isinstance(arr[mid][index], datetime):
            mid_value = arr[mid][index]
        elif isinstance(arr[mid][index], date):  # Handle datetime.date as well
            mid_value = datetime(arr[mid][index].year, arr[mid][index].month, arr[mid][index].day)
        else:
            mid_value = datetime.strptime(arr[mid][index], '%Y-%m-%d')

    elif index == 'capacity':
        mid_value = int(arr[mid][index])

    else:  # target_time
        mid_value = arr[mid][index]

    # Binary Search comparisons
    if mid_value < key:
        low = mid + 1
    else:
        high = mid - 1

return low
```

- Efficient for finding venues in sorted data.
- Best/Average/Worst = O(1)/O(log n)/O(log n).

Sorting Algorithm

Back to Agenda Page

Merge Sort

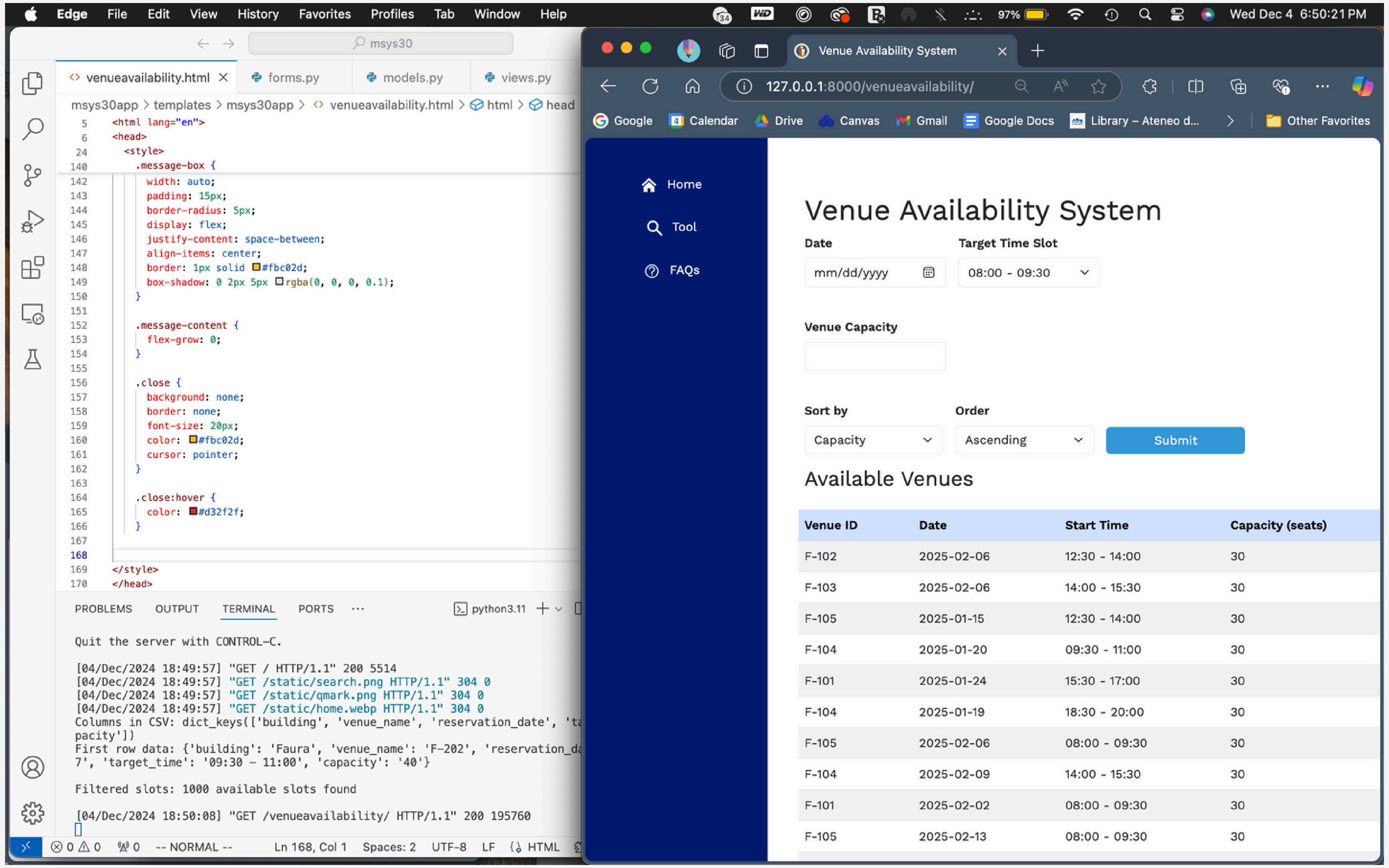


```python
def merge_sort(data, key):
    """
    Merge sort to sort a list of dictionaries by a specific key given as a parameter.
    """
    if len(data) <= 1:
        return data

    mid = len(data) // 2
    left = merge_sort(data[:mid], key)
    right = merge_sort(data[mid:], key)

    return merge(left, right, key)

def merge(left, right, key):
    """
    Merge two sorted lists.
    """
    merged = []
    i, j = 0, 0
    while i < len(left) and j < len(right):
        if left[i][key] <= right[j][key]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            j += 1

    merged += left[i:]
    merged += right[j:]
    return merged
```

- Stable and efficient for handling large datasets.
- Ensures consistent performance for future scalability.
- **Time Complexity:**
    - O(n log n).

MSYS 30 Final Project

Demo with Full Code



```
<html lang="en">
<head>
  <style>
    .message-box {
      width: auto;
      padding: 15px;
      border-radius: 5px;
      display: flex;
      justify-content: space-between;
      align-items: center;
      border: 1px solid #fbc02d;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }

    .message-content {
      flex-grow: 0;
    }

    .close {
      background: none;
      border: none;
      font-size: 20px;
      color: #fbc02d;
      cursor: pointer;
    }

    .close:hover {
      color: #d32f2f;
    }


</style>
</head>
```

```
Quit the server with CONTROL-C.

[04/Dec/2024 18:49:57] "GET / HTTP/1.1" 200 5514
[04/Dec/2024 18:49:57] "GET /static/search.png HTTP/1.1" 304 0
[04/Dec/2024 18:49:57] "GET /static/qmark.png HTTP/1.1" 304 0
[04/Dec/2024 18:49:57] "GET /static/home.webp HTTP/1.1" 304 0
Columns in CSV: dict_keys(['building', 'venue_name', 'reservation_date', 'ta
pacity'])
First row data: {'building': 'Faura', 'venue_name': 'F-202', 'reservation_da
7', 'target_time': '09:30 - 11:00', 'capacity': '40'}

Filtered slots: 1000 available slots found

[04/Dec/2024 18:50:08] "GET /venueavailability/ HTTP/1.1" 200 195760
```

Venue Availability System

Date: mm/dd/yyyy

Target Time Slot: 08:00 - 09:30

Venue Capacity:

Sort by: Capacity

Order: Ascending

Available Venues

Venue ID	Date	Start Time	Capacity (seats)
F-102	2025-02-06	12:30 - 14:00	30
F-103	2025-02-06	14:00 - 15:30	30
F-105	2025-01-15	12:30 - 14:00	30
F-104	2025-01-20	09:30 - 11:00	30
F-101	2025-01-24	15:30 - 17:00	30
F-104	2025-01-19	18:30 - 20:00	30
F-105	2025-02-06	08:00 - 09:30	30
F-104	2025-02-09	14:00 - 15:30	30
F-101	2025-02-02	08:00 - 09:30	30
F-105	2025-02-13	08:00 - 09:30	30

MSYS 30 Final Project

Edge Cases

Scenario	Message Type	Message Content
Data outside 01-15-2025 to 02-14-2025 range	Warning	Venue availability data is not available for this date range.
No results after filtering	Warning	No match found.
Results available after filtering	Success	Filter complete. Available venues displayed below.
No data in load_reservations_data()	Error	No reservation data found.

Results and Evaluation

- **Accuracy**: Correct venues displayed based on search criteria.
- **Efficiency**: Sorting within O(n log n) and searching in O(log n).
- **Robustness**: Handles unexpected inputs without crashing.
    - Form Data Validation
    - Able to handle the following cases:
        - Data outside Allowable Date Range
        - No results
        - No data
- **Performance Goals**: Processes small datasets in under 1 second.

Future Directions

- Expand scope to other buildings.
- Integration with Real-Time Data
- Incorporation of the Booking Process

Venue Availability System

Streamlining Venue Reservations for Ateneo CFMO

Jana Almira Boco, Nathan Luna, Juliana Valdez