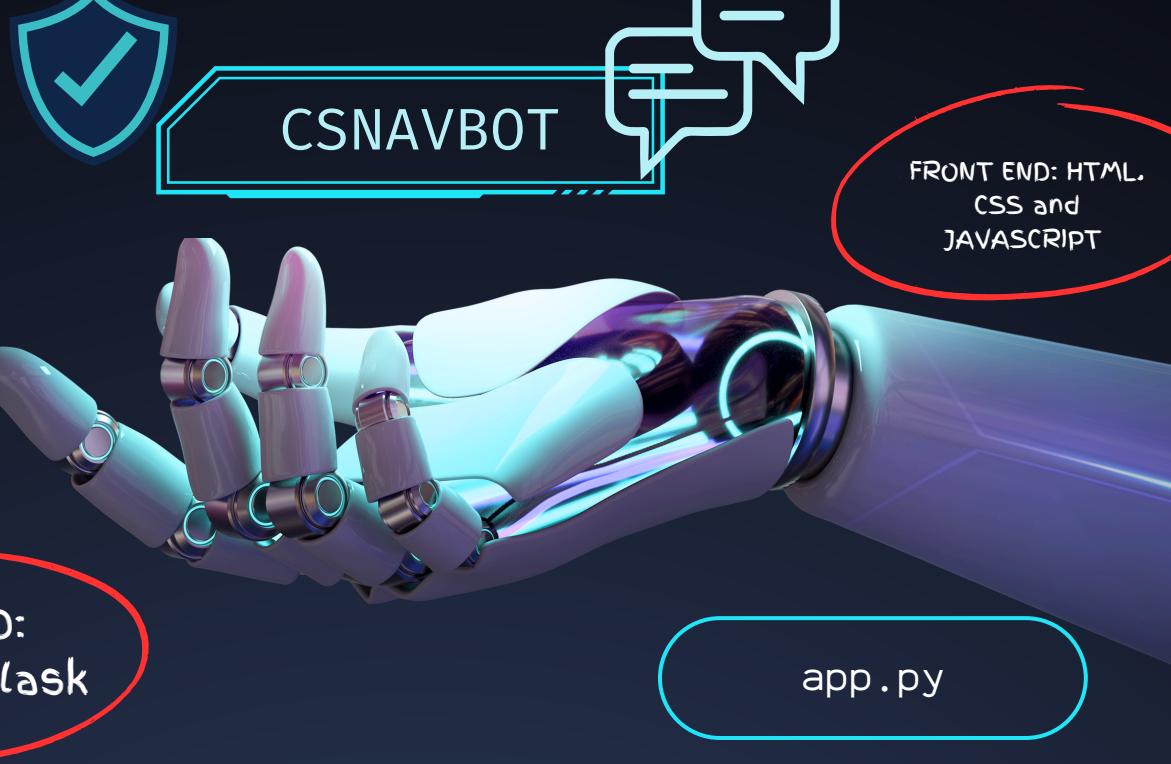


WID3002

NATURAL LANGUAGE PROCESSING



Defines a Flask web application that provides a chatbot-like interface for assisting users with navigation, events, and courses at a university.

1	home	The main route that renders the homepage with map, event, and course data.
2	read_map_data, read_event_data, read_course_data	Reads and processes map, event and course data from a Google Sheets document.
3	location_exists, event_exists	Checks if a given location/event exists in the map data.
4	get_nav_image	Generates a navigation image based on the user's start and destination points.
5	get_stored_data	Returns stored navigation answers and current session data.
6	get_bot_response	Processes user input and returns a bot response for navigation help.

pngcsv.py

loads an image of a floor plan, preprocesses it, and then extracts the pixel data to create a simplified representation of the floor plan in a CSV

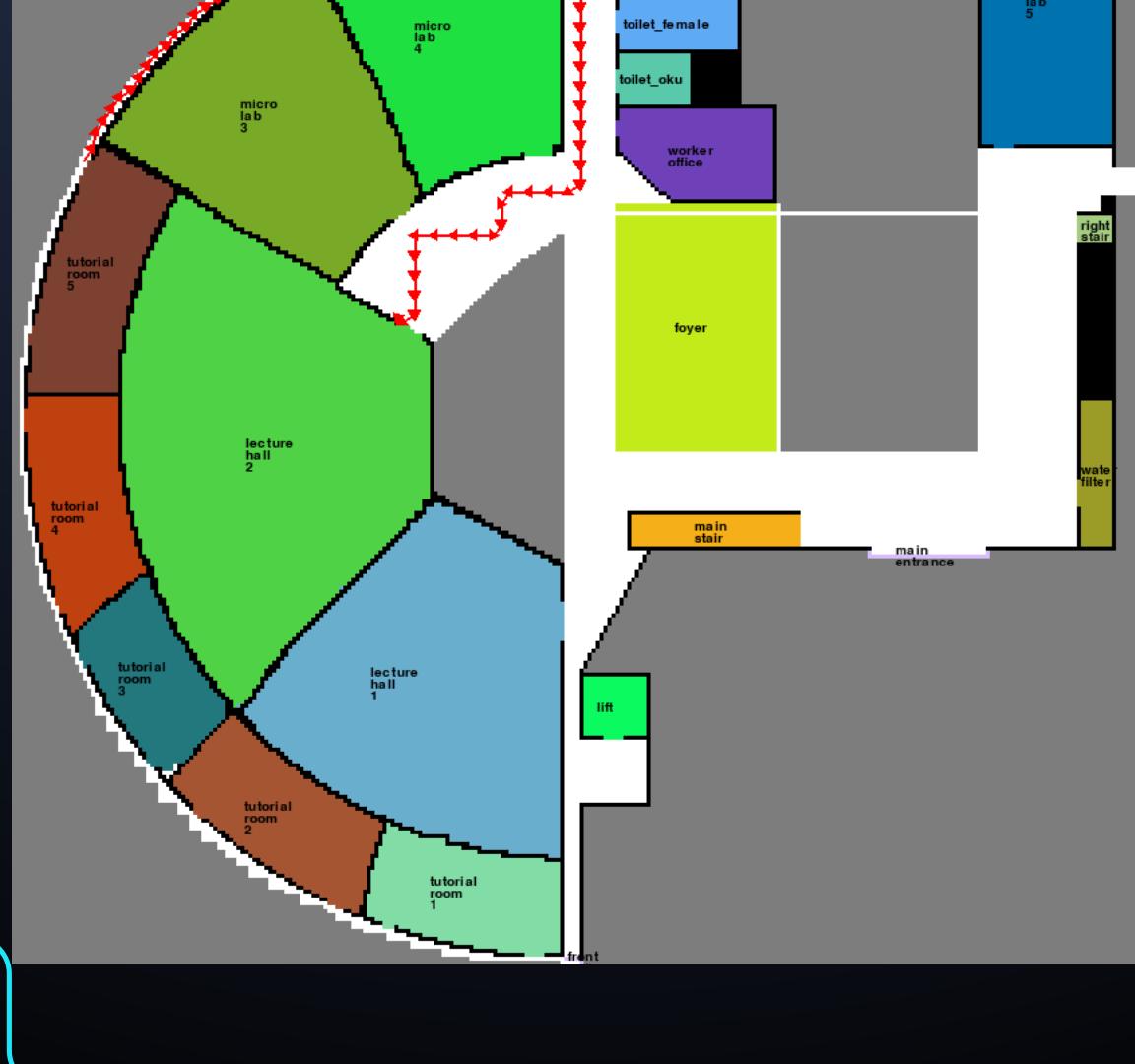
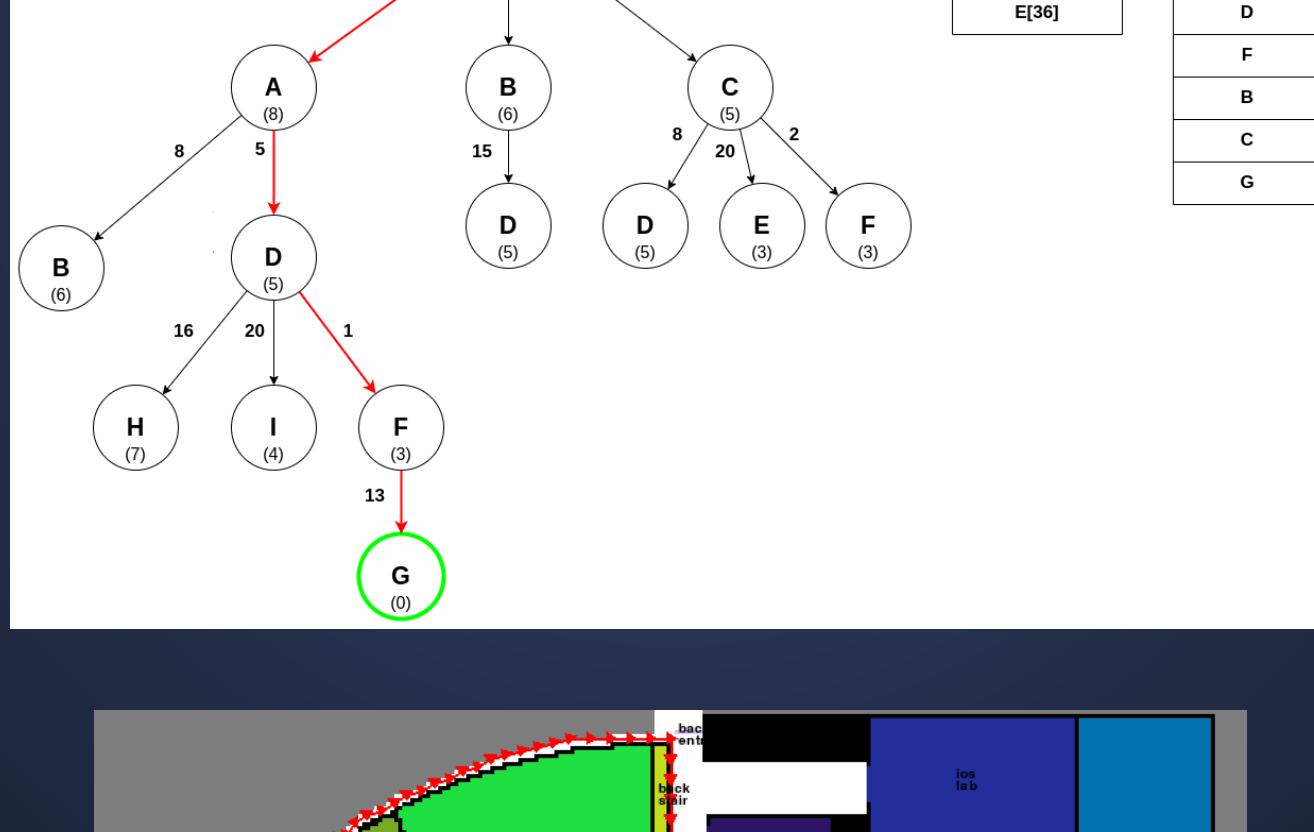
1	Image Loading	The script loads an image from a specified path and resizes it.
2	Convert Grayscale	Converts the image to grayscale.
3	Thresholding	Applies a binary threshold to separate the image content from the background.
4	Contour Detection	Finds contours in the thresholded image.
5	Data Conversion	Converts the image data to a NumPy array, modifies it by changing specific pixel values, and removes columns containing only zeros.
6	CSV File Creation	Writes the processed pixel data into a CSV file

A* ALGORITHM

nav_algo.py

Contains functions related to pathfinding using the A* algorithm for navigation in a maze-like environment represented by CSV files.

1	A* Algorithm	It includes functions to calculate the distance and heuristic between points, and the main A* function for pathfinding.
2	Floor Transitions	The code handles transitions between different floors, which is useful for multi-level pathfinding scenarios.
3	Accessibility	There's a consideration for users with disabilities (is_OKU parameter), ensuring the path includes lifts where necessary.
4	Dynamic Path Drawing	Utilizes the pygame library to draw the path on a maze representation, which could be part of a graphical user interface.

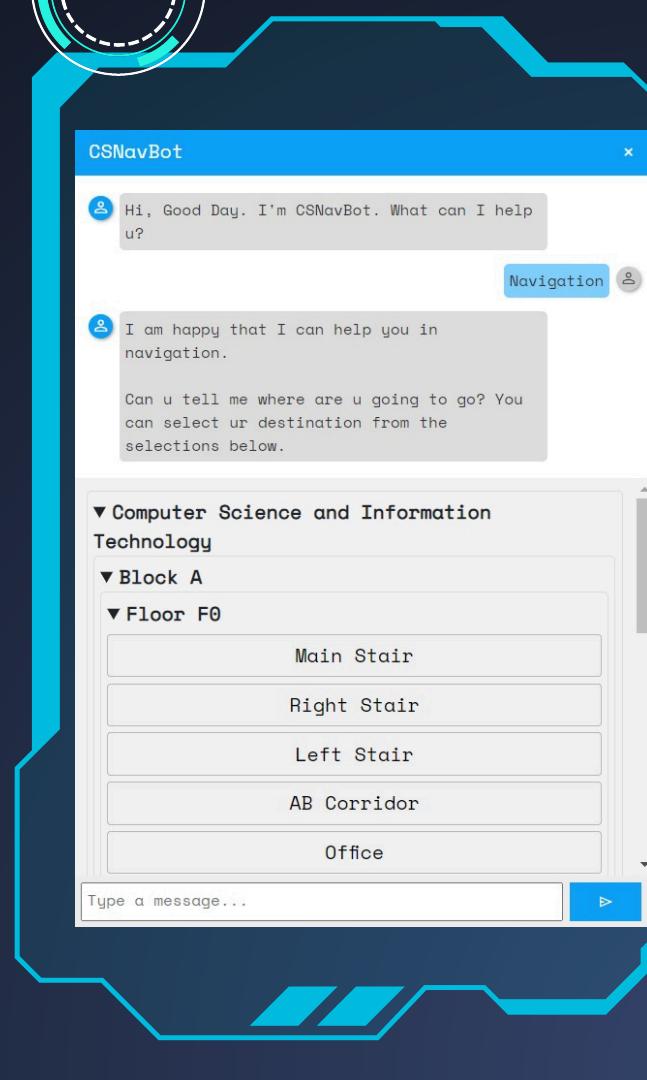




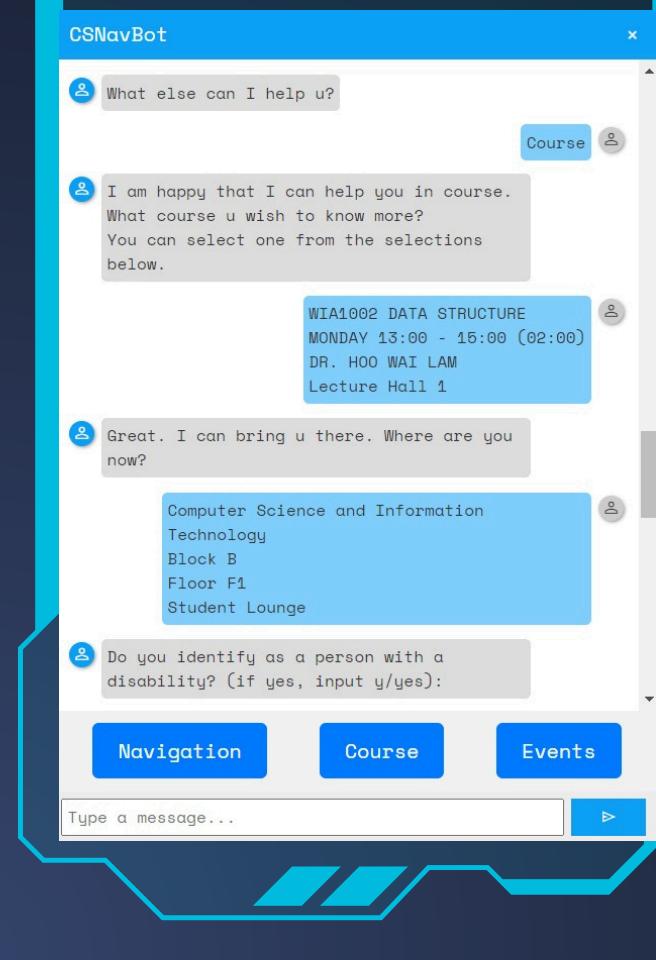
DATASETS

- **Room Database**
 - a. Contains information about the rooms/facilities, including location, names and so on
- **Transition Database**
 - a. Contains data on how to transition between different floor
- **STU_MVT4**
 - a. Data get from Maya about the courses
- **Event Database**
 - a. Include schedules or occurrences in UM
- **Map Data**
 - map for different floors or buildings
 - a. **AGF**
 - b. **AF1**
 - c. **AF2**
 - d. **BGF**
 - e. **BF1**
 - f. **BF2**

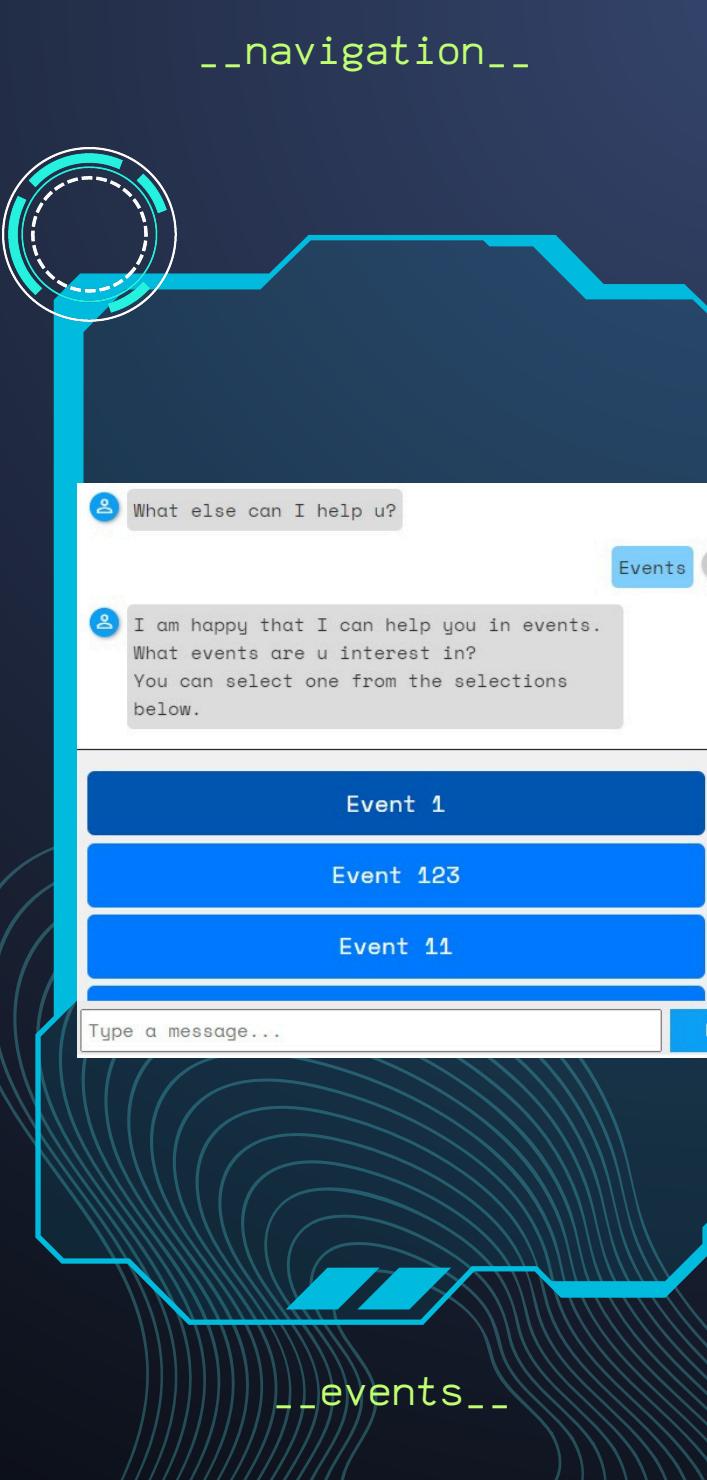
DEMO



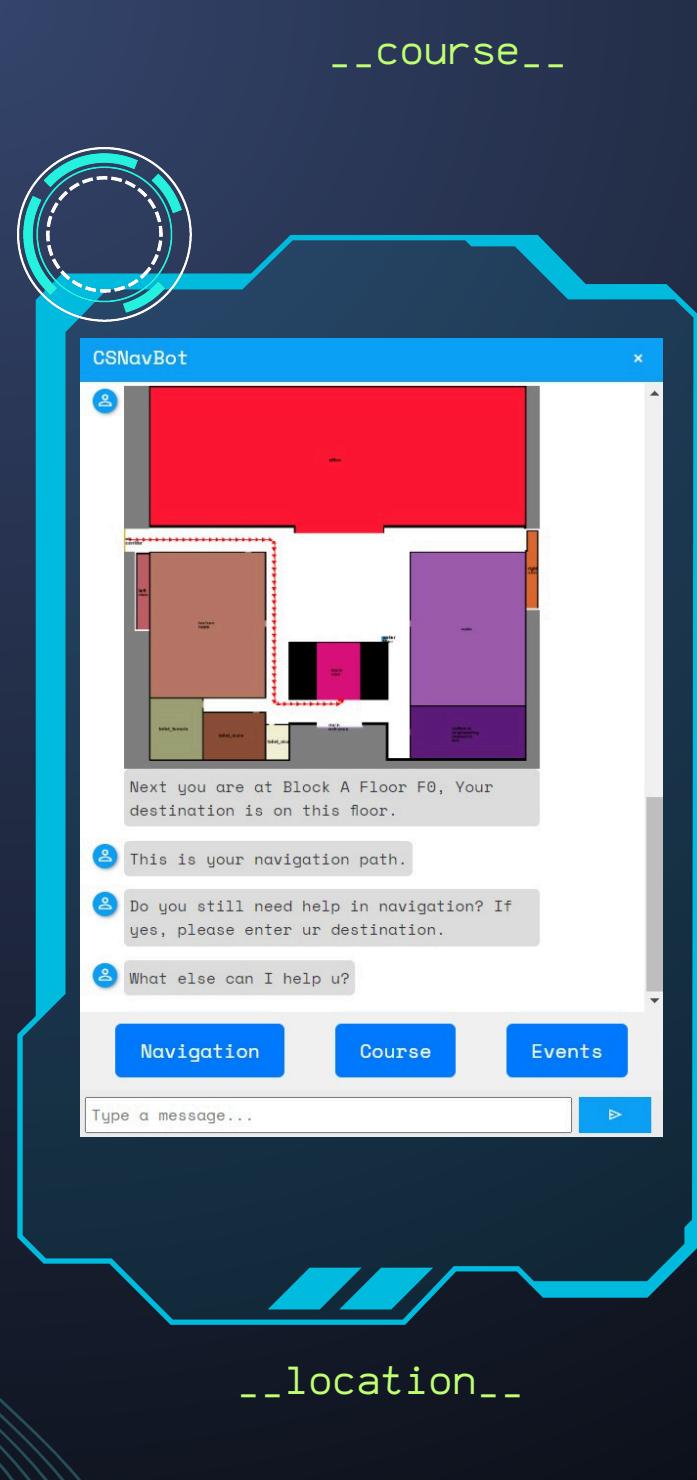
--navigation--



--course--



--events--



--location--

Prepared by:

- Lai Ke Wei
- Reem Idres
- Ibtism Gul
- MD Raffaul Islam
- Azizar Mohammad Sdmam Sobhan
- Surayia Rahman

