Group 1
Joaquin Garcia & Linn Hjulstad

Garden Paradise

Table of contents

Users, main functions and tools we used

01

Introduction

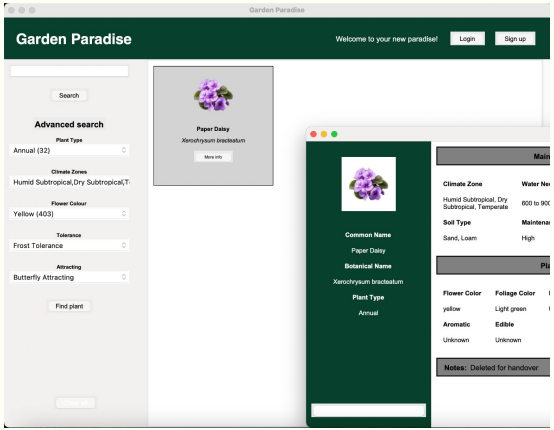
Users

New plant owners

A plant search app

- Plant owners who want to expand their plant collection
- Have trouble finding out which plant to buy
- Help narrow down choices

Main functions

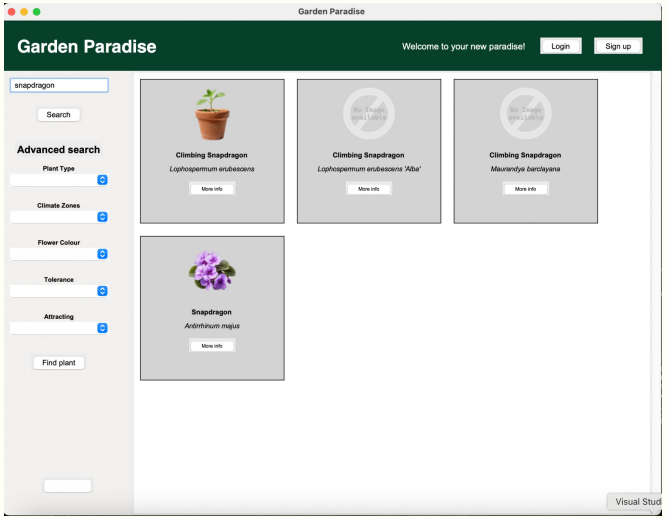


Plant search

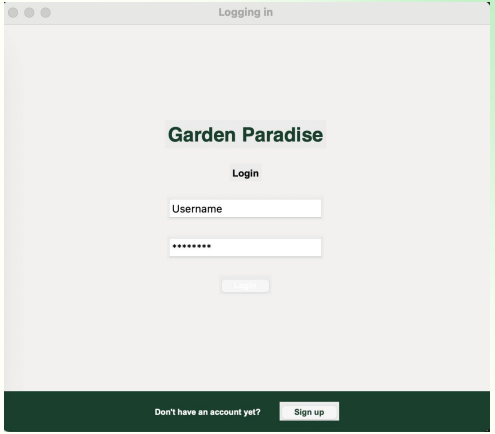
Two search functions:

- **Quick search**, or
- **Advanced search** based on plant characteristics
- Plant results show up on screen
- Limited to 50 results

- Initially we also wanted a top bar for users to sort plants by maintenance, height ranges and spread ranges
- Would be helpful especially for new plant owners
- With time constraints we made a compromise

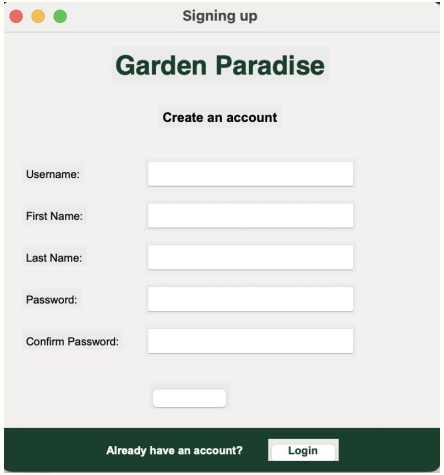


Main functions 2



Login and sign up

- Can make an account attached to your plant search
- Add plants to **favorites** list
- Or delete them



Languages, frameworks and packages

Language

- Python
- SQL

Frameworks

- Tkinter for GUI development

Packages

- Pillow (PIL) for opening and resizing images

02

DB design and modelling

Our data

Plants dataset

- Waterwise plants dataset from Queensland Government
- https://www.data.qld.gov.au/dataset/waterwise-plants/resource/fd297d03-bf72-40c7-b27e-24cc7023360c
- CSV file with 4654 records
- From Australia, so many exotic plants, not just houseplants
- Images

Data cleaning and import methods

- For some reason had two ID columns
- Removed columns related to images
- Made our own image table
- Otherwise we would've had to do website scraping with around 500 website links
- Imported data in MySQL workbench with the Import Wizard

_id	Plant ID	Plant Co…	Botanic…	Commo…	Previou…
1	2	ABECHI	Abelia ch…	Chinese …	
2	3	ABEFLO	Abelia flo…	Mexican …	
3	4	ABEGRA	Abelia x …	Abelia	
4	5	ABEFRA	Abelia x …	Golden …	
5	6	ABEGOL	Abelia x …	Golden …	
6	7	ABEESC	Abelmos…	Okra	
7	8	ABEMAN	Abelmos…	Abika	
8	9	ABEPAL	Abelmos…	Palm Le…	
9	10	ABEMOS	Abelmos…	Musk Ma…	
10	11	ABEMIS	Abelmos…	Rose Mu…	
11	12	ABEEVE	Abelmos…	Rose Mu…	
12	13	ABIPRO	Abies pr…	Noble Fir	
13	14	ABRANG	Abroma …	Devil's C…	
14	15	ABUMEG	Abutilon …	Big River…	
15	16	ABUVAR	Abutilon …	Variegat…	
16	17	ABUOXY	Abutilon …	Small Fl…	
17	18	ABUBOU	Abutilon …	White Ab…	
18	19	ABUGOL	Abutilon …	Yellow A…	
19	20	ABUORA	Abutilon …	Orange …	
20	21	ABURED	Abutilon …	Red Abu…	
21	22	ABUSAV	Abutilon …	Variegat…	
22	23	ABUSOU	Abutilon …	Variegat…	

Making images table

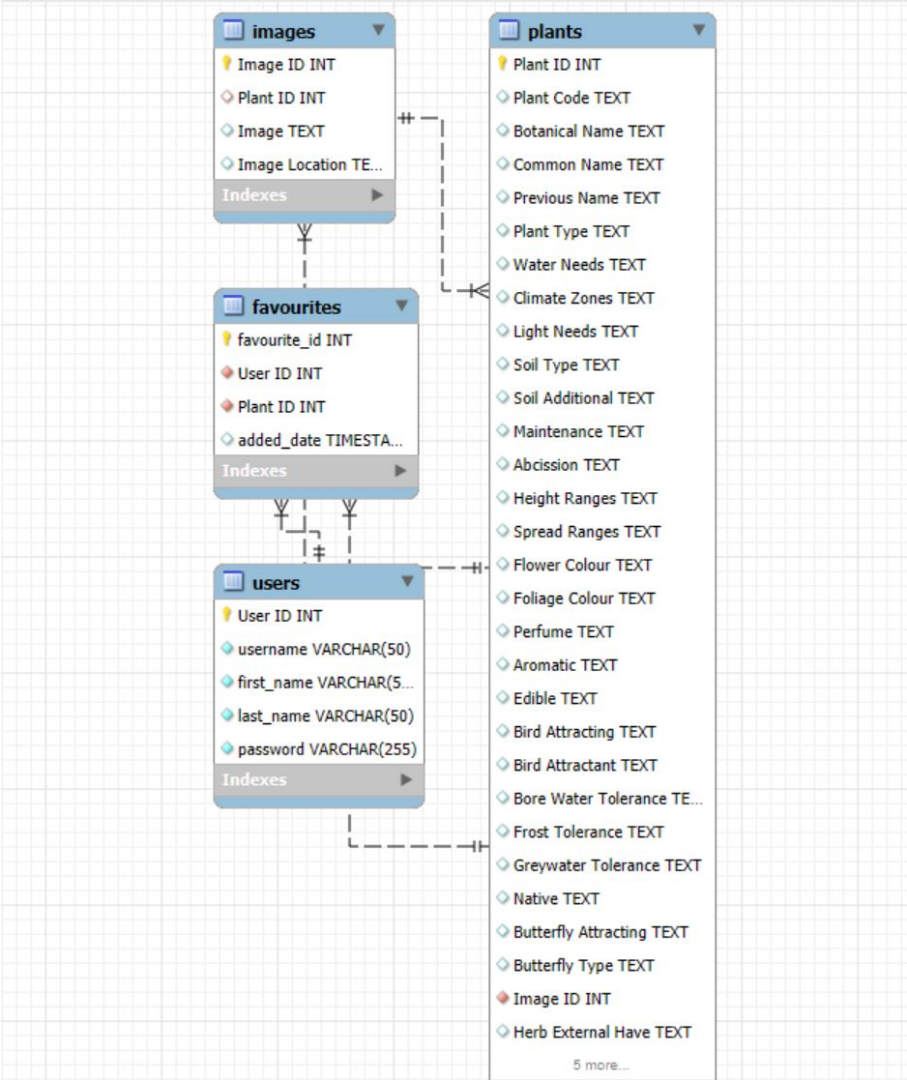
- Downloaded around 40 images manually from Vecteezy.com
- A default image for approximately each plant type
- Some flower colour, such as annual plants and orchids
- Replaced the old image paths with new ones with UPDATE queries


```sql
#Shrub default
UPDATE images
SET `Image Location` = 'plants/shrub/shrub_default.png'
WHERE `Image` LIKE "Yes"
AND `Plant ID` IN (
    SELECT `Plant ID`
    FROM plants
    WHERE `Plant Type` LIKE "%Shrub%"
);

#Shrub, vegetable
UPDATE images
SET `Image Location` = 'plants/shrub/shrub_vegetables.png'
WHERE `Image` LIKE "Yes"
AND `Plant ID` IN (
    SELECT `Plant ID`
    FROM plants
    WHERE `Plant Type` LIKE "%Shrub%"
    AND `Plant Type` LIKE "%vegetable%"
);
```

DB model and ER diagram

- Primary keys
- User ID
- Plant ID
- Favourites



Interface design, GUI managers, components

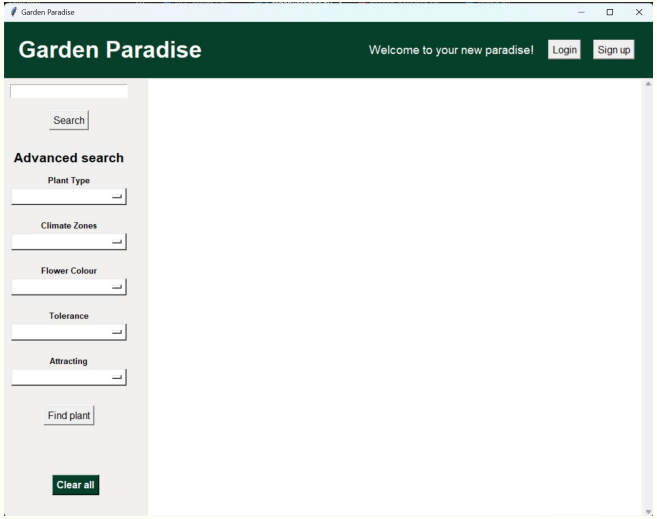
03

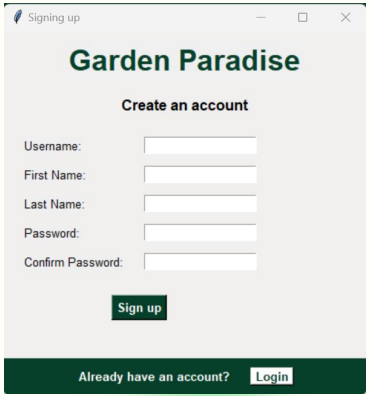
GUI design

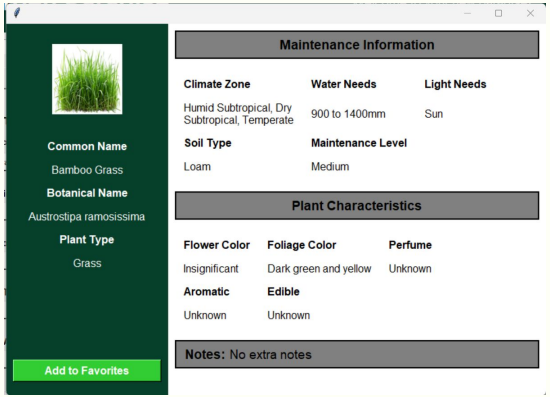
GUI design

We used a combination of **grid**, **pack** and **place** as layout managers

- Dropdown menus
- Entry
- Buttons





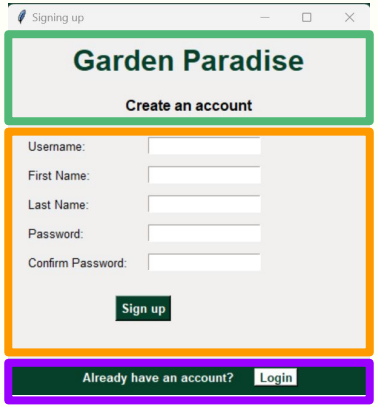


Layout Manager Combination - Ex.

Title Frame =
- Using pack to center

Main Frame
- Using grid for control

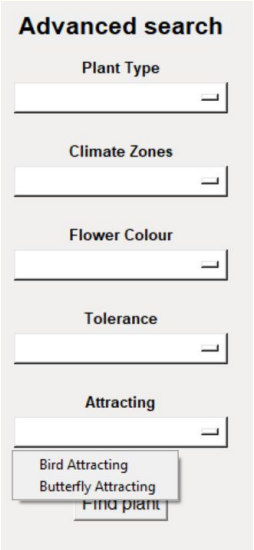
Bottom Frame
- Using pack to center

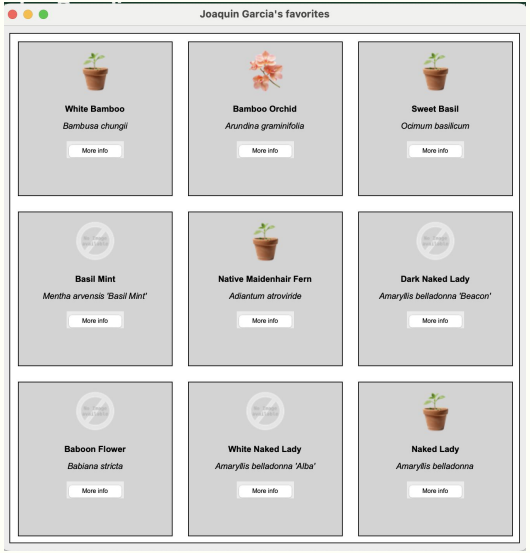


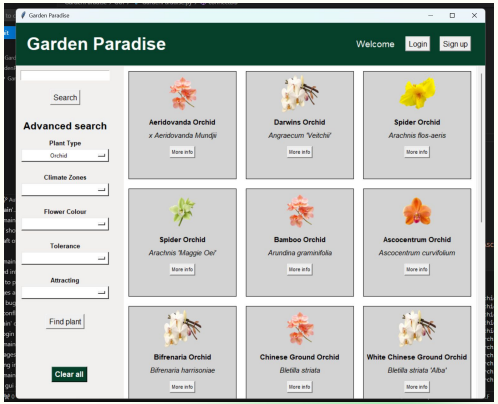
User Interactions

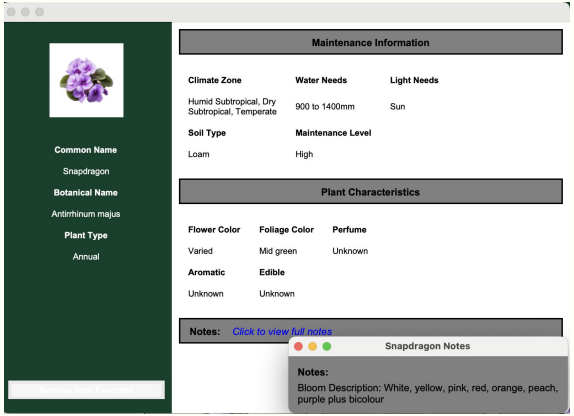
For the avid plant finder:

- Search by characteristics
- Search by name (common and botanical)
- Compact info window
- Simple saved favourites





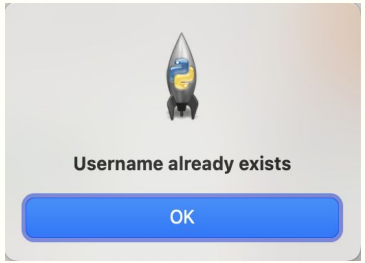


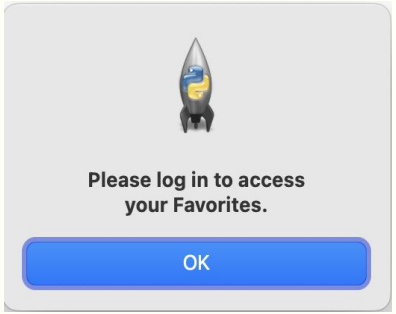


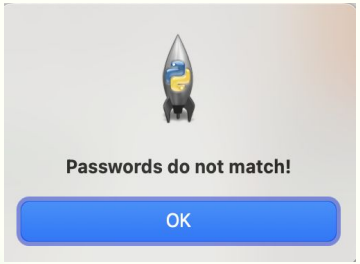
Possible User Errors

Mostly through **Database Interactions**

- Incomplete sign-up
- Existing username
- Unmatched passwords
- Access to saved favourites







04

Demo

Demo time!

SELECT, INSERT, DELETE, JOIN, GROUP BY

05

SQL statements

SQL statements

```
query = "INSERT INTO favourites (`User ID`, `Plant ID`) VALUES (%s, %s);"
```

```
query = """
SELECT `Common Name`, `Botanical Name`, `Plant ID`
FROM plants
WHERE `Common Name` LIKE %s
OR `Botanical Name` LIKE %s
OR `Plant Type` LIKE %s
ORDER BY `Common Name` ASC
LIMIT 50;
"""
```

```
query = f"""
    SELECT {column_name}, COUNT(*)
    FROM plants
    GROUP BY {column_name}
    ORDER BY {column_name} ASC
"""
```

```
("DELETE FROM favourites WHERE `User ID` = %s AND `Plant ID` = %s"
```

```
query = """
SELECT img.`Image Location`
FROM images img
JOIN plants p ON p.`Image ID` = img.`Image ID`
WHERE p.`Plant ID` LIKE %s
"""
```

```
query = """
SELECT p.`Common Name`, p.`Botanical Name`, p.`Plant ID`
FROM favourites f
JOIN plants p ON f.`Plant ID` = p.`Plant ID`
WHERE f.`User ID` = %s ORDER BY f.added_date DESC LIMIT 9;
"""
```

06

Lessons learned

Challenges and lessons learned

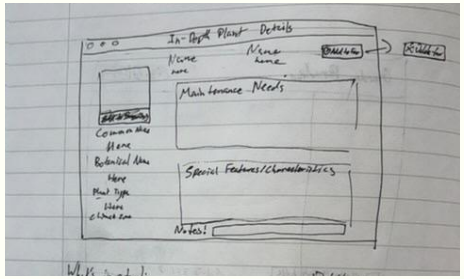
01	02	03	04	05
Import and data cleaning sooner rather than later	Collaboration and sharing databases – mixed column names across tables. `User ID` vs. favourite_id	UI designs – time-consuming	File organization	Linking app components together
06	**07**	**08**	**09**	**10**
Multiple frames and GUI organizers	Importing images – web scraping	Different environments (macOS vs WindowsOS)	Database configuration (config.py and .gitignore)	Time constraints

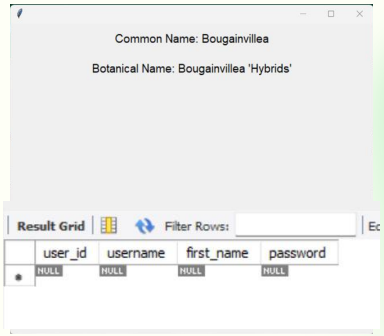
07

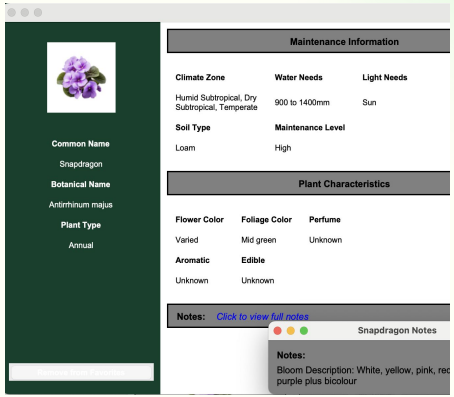
Timeline

Timeline

- Database import and cleaning
- Plants and image table creation
- Main screen
- Users and favorites table creation
- Login and sign up
- Favorites + App Integration
- Adding images







Ideation Phase

Development Phase

Launch Phase

08

Q&A? Thank you!

Appendix

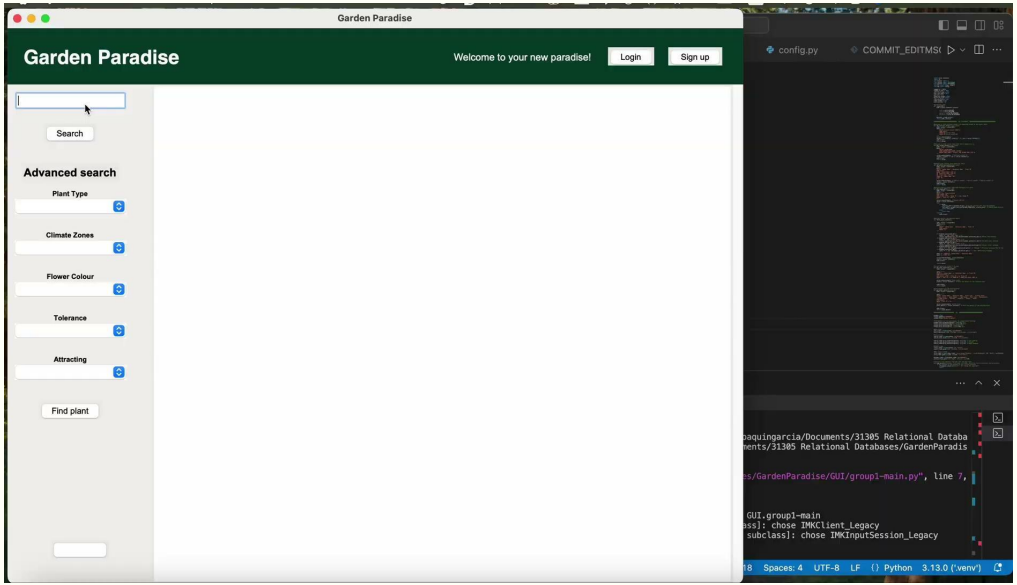
NEW (Update): Favourites

More “Favourites” and unique window robustness

- Favourites table updates whenever the favourites button is toggled
- The favourites button is only accessible when the user is logged in (refreshed on log-in and log-out)
- During log-in and log-out, all relevant active windows are refreshed (“favorites”, all active “plant_windows”)

Warning to professor:

- Not sure if there’s something wrong with my VS Code again, but Joaquin can’t run the main file except using **python -m GUI.group1-main** otherwise, works fine



NEW: Updated Functions to Support Favourites

toggle_favorites(plant_id, user_id, plant_window)

check_if_favorited(user_id, plant_id)
log_out()

open_login_screen()

validate_login(username, password, login_window)

create_favorite_button(left_frame)

update_favorite_button(plant_window, user_id, plant_id)

open_sign_up_screen()

sign_up_user(username, first_name, last_name, password, confirm_password, sign_up_window)

show_selected_plant(plant_id, user_id)

refresh_favorites()

NEW: Global Variables

Main reasoning for global variables

logged_in = FALSE
- menu buttons, toggle-favorites buttons

logged_in_user = None
- menu buttons, window names

user_first_name = None
- window names, welcome message

user_last_name = None
- window names, welcome message

user_id = None
- window names, current user, favourites

favorites_window = None
- only one active at a time, can refresh

favorites_frame = None
- for refresh purposes

sign_up_window = None
- only one active at a time

login_window = None
- only one active at a time

plant_windows = {}
- controls plant window references

```
logged_in = FALSE
logged_in_user = None
user_first_name = None
user_last_name = None
user_id = None
favorites_window = None
favorites_frame = None
sign_up_window = None
login_window = None
plant_windows = {}
```