EcoPlot Design Doc

EcoPlot is an application designed for gardeners who want to thoughtfully design their garden especially for those tight on space. The application includes a database of fruits and vegetables and some information including spacing, soil, water and sun requirements. You can search for plants based on your garden's specs and if you give garden dimensions you can layout the potential plants you'd like to include. The garden design process includes interactive plant sizing allowing you to see the actual layout to scale of your garden.

Application State and Flow

The application allows users to:

- Search for plants based on their garden specs.
- View suggested plants.
- Drag and drop selected plants onto a garden grid
- Undo plant placements or clear gardens.

Frontend

- PlantSearch.jsx
    - plantDetails: Tracks garden conditions imputed by users.
    - gardenSuggestions: List of suggestions after their gotten
    - expandedPlant: Tracks which plant's details are expanded in the suggestion list.
- GardenGrid.jsx:
    - plants: Current plants on the grid, each plant has an ID, coordinates, spacing and a name.
    - plantHistory: History of plant placements for undo functionality.
    - cellSize: Changing size of each grid cell, based on the size of the grid container.
  - App.js:
    - gardenSize: Holds the user-selected garden dimensions (length x width).

Backend (Flask)

- main.py:
    - Hosts the Flask app and endpoint blueprints.
- routes/plants.py
    - Defines the endpoints called in the frontend.
- model/plants.py
    - Deals with all the database interactions.

- db.py:
    - Holds database connection

Data Flow

1. User selects garden conditions ->
2. React gives the conditions to Flask ->
3. Flask returns plant suggestions ->
4. User drags and drops suggestions onto the grid ->
5. Plant placement is recorded and visually shown.

Why This Application State and Flow?

The general design of the project directory with every component separated from the frontend to the models and routes allows for easy access. With everything in its own folder it's very intuitive to find a desired component, this layout also allows for easy expansion without things getting cluttered.

In regards to the frontend there were a few meaningful decisions I made that I think elevated the project. Creating a dynamic cell size allows for the grid to always take up a reasonable space while also allowing for a to scale garden representation. This was important for incorporating the spacing requirements of the plants and creating varying sizing based on this. Continuing on this aspect, the dynamic sizing not only adds depth to the project but turns what could be a cool toy into a useful gardening tool. The last feature that turned out to be really important was the undo and clear button as one time placement is not very feasible.

The backend is where I feel my work was the strongest, using Flask a tool I'm comfortable with allowed for creating a very lightweight backend. The database model contains all the functions working with the database most of which are pretty straightforward. The routes just call these functions allowing them to interact with the frontend. The one function that is more than just moving data is the recommendation function. This function is still pretty simplistic matching the specs given by the user to specs of plants in the database, for each matching spec the score of that plant is increased with the plants with the top 15 scores returned.

Honestly considering the accessibility of my application post production, there is a bit to be desired. I added some mouse changing to the pointer and used a color palette to keep things looking clean. The reason I say there are some accessibility features

missing is since there weren't any conscious decisions made in the name of accessibility. I could probably go back in and add some features that could fix this and make my application more accessible.

Overall this project was very fun to create especially since the final product resulted in a functioning tool that I foresee my self using in the future. The application accomplishes the goal of allowing for an easy way to layout your garden taking all the things you might consider and assisting you with companion planting suggestions.