**GrowTime**

*Project Plan*

**CONCEPT**

Problem: various and disorganized sources of information on when to plant, fertilize, spray, prune, and otherwise care for outdoor garden plants

Solution: collects and displays information in a user-friendly calendar, programmed to send text reminders

Target User: amateur gardeners in planting zones 4-6

**FEATURES**

MVP:

* login page to track user sessions & create a customizable experience
* master calendar that displays gardening tasks by month
* list of a user’s current plants & optional plants to add to their collection
* plant profile pages, displaying the stats & task dates for each plant
* double-click feature on plant profiles to allow for name/date changes
* form allowing for the input of a new plant into the database
* automated text reminder for each task

Additions:

* limit access for double-click edit feature & new plant form to admins
* link to instructional video for each task (included in text)
* printable calendar
* drag-and-drop calendar with customizable events
* optional email reminders
* extended plant database (trees & shrubs, extended zones, etc.)
* extended plant profile (growing information, gardening tips, etc.)

**VIEWS**

**Logo**

It’s… GrowTime!

*You grow it. We’ll time it.*

Username

Password

**Login**

**REGISTER** as a new user.

Login:

Register:

**Logo**

First Name

**Register**

Last Name

Email

City

Phone (for text notifications)

Password

State

Username

**Logo**

**June**

**Calendar**

**Add a Plant**

**Plant Profiles**

**Plant List**

Calendar:

**Logo**

**Plant List:**

*Click to add or remove a plant from your calendar.*

*Double-click to visit the plant profile.*

**Calendar**

**Add a Plant**

**Plant Profiles**

**Plant List**



Azalea

Bleeding Heart

Burning Bush

Black-eyed Susan

**Search:**

Enter Common Plant Name

Plant List:

First Name

Plant Profile:

**Logo**

**Bleeding Heart**

*Lamprocapnos spectabilis*

**Calendar**

**Add a Plant**

**Plant Profiles**

**Plant List**



**Calendar Items**

Plant: April 15th

Fertilize: April 15th

Bloom: May 1st

Treat: May 15th – Snails

June 1st – Snails

Spent: June 20th

Prune: July 1st

Hardiness: Zone 3-9

Exposure: Shade/Part Sun

**Edit**

**Delete**

**<**

**>**

Add a Plant:

**Logo**

**Calendar**

**Add a Plant**

**Plant Profiles**

**Plant List**

**Calendar Items**

Plant: |

Fertilize: | – |

Bloom: |

Treat: | – |

| – |

Spent: |

Prune: |

Other: |

Photo URL: |

Hardiness: |

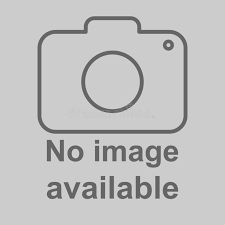
Exposure: |

**Delete**

**Save**

**Common Name**

*Scientific Name*



**ROUTES**

* Login: ‘/’
* Calendar: ‘/calendar’
* Plant List: ‘/plant-list’
* Plant Profile: ‘/plant-profile-1’ (etc.)
* Add a Plant: ‘/add-a-plant’

**DATABASE SCHEMA**

create table growtime\_users (

id serial primary key,

username varchar(25),

password text

);

create table growtime\_profiles (

user\_id integer references growtime\_users(id),

first\_name varchar(20),

last\_name varchar(30),

email varchar(40),

phone varchar(10),

city varchar(20),

state varchar(20)

);

create table plants (

plant\_id serial primary key,

common\_name varchar(40),

scientific\_name varchar(60),

plant\_pic text,

description varchar(200),

hardiness varchar(15),

exposure varchar(20)

);

create table plant\_dates (

plant\_id integer references plants(plant\_id),

planting\_date date,

fertilize\_date\_1 date,

fertilize\_date\_2 date,

fertilize\_date\_3 date,

bloom\_date date,

treatment\_date\_1 date,

treatment\_type\_1 varchar(20),

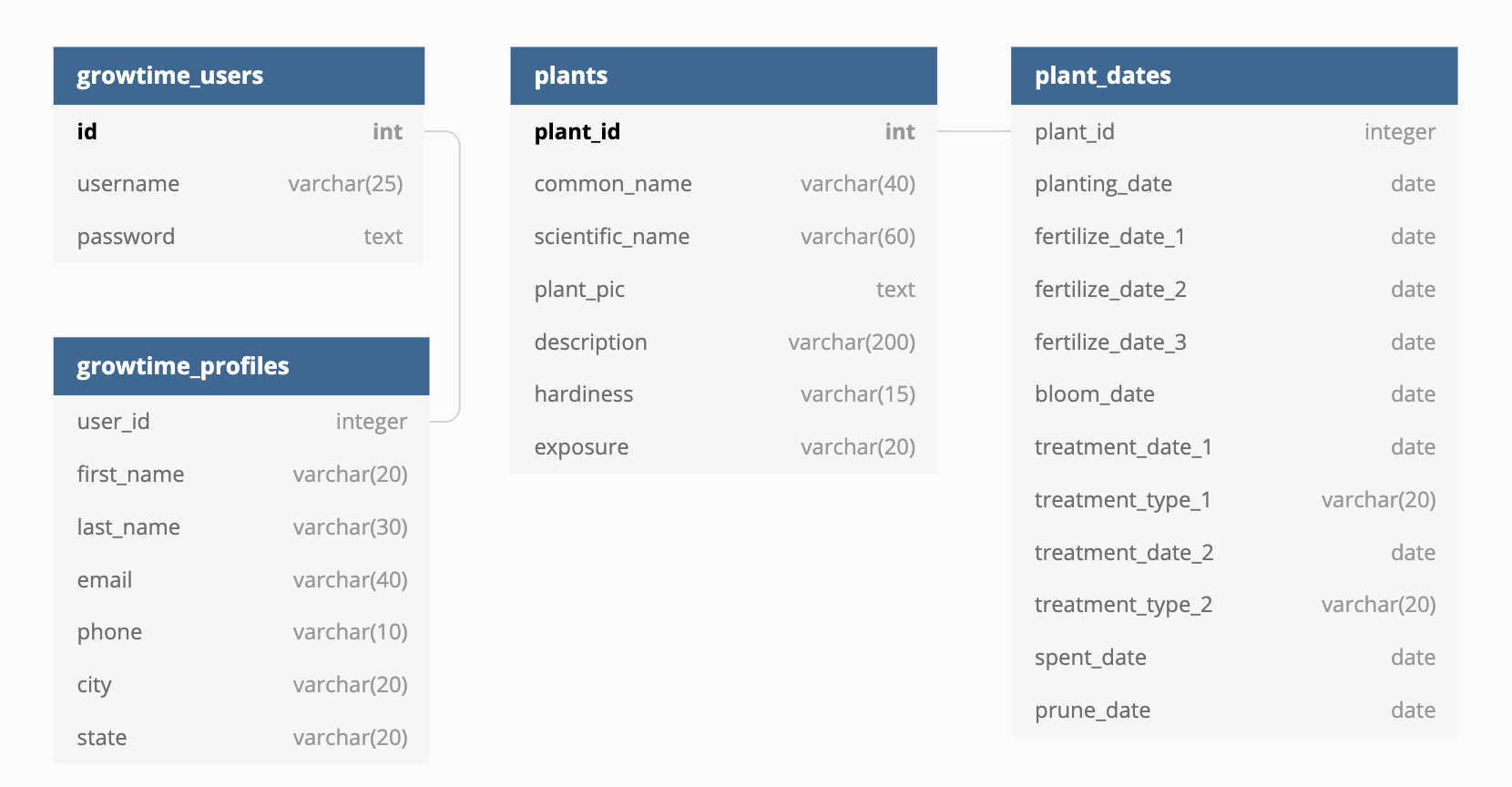
treatment\_date\_2 date,

treatment\_type\_2 varchar(20),

spent\_date date,

prune\_date date

);



**ENDPOINTS & CONTROLLERS**

authCtrl:

* app.post (‘/auth/register’, authCtrl.**register**)
  + input collected on body & saved to the following SQL tables:
    - growtime\_users: username, password – generates id
    - growtime\_profiles: first\_name, last\_name, email, phone, city, state
* app.post (‘/auth/login’, authCtrl.**login**)
  + input collected on body and authenticated against SQL table:
    - growtime\_users: username, password

plantCtrl:

* app.get (‘/api/plants’, plantCtrl.**getPlants**)
  + query by common plant name, else return all
  + map over collection and return from SQL:
    - plants: plant\_pic, common\_name
* app.get (‘/api/plant/:plantid’, plantCtrl.**getPlant**)
  + locate plant by plantid param
  + map over collection and return profile cards from SQL:
    - plants: common\_name, scientific\_name, plant\_pic, description, hardiness, exposure
    - plant\_dates: planting\_date, fertilize\_date\_1, fertilize\_date\_2, fertilize\_date\_3, bloom\_date, treatment\_date\_1, treatment\_type\_1, treatment\_date\_2, treatment\_type\_2, spent\_date, prune\_date
* app.post (‘/api/newplant’, plantCtrl.**newPlant**)
  + input collected on body & saved to the following SQL tables:
    - plants: common\_name, scientific\_name, plant\_pic, description, hardiness, exposure
    - plant\_dates: planting\_date, fertilize\_date\_1, fertilize\_date\_2, fertilize\_date\_3, bloom\_date, treatment\_date\_1, treatment\_type\_1, treatment\_date\_2, treatment\_type\_2, spent\_date, prune\_date
* app.delete (‘/api/plant/:plantid’, plantCtrl.**deletePlant**)
  + locate plant by plantid param and delete from the following SQL tables:
    - plants
    - plant\_dates
* app.put (‘/api/plant/:plantid’, plantCtrl.**editPlant**)
  + input collected on body then saved as changes on SQL tables:
    - plants: common\_name, scientific\_name, plant\_pic, description, hardiness, exposure
    - plant\_dates: planting\_date, fertilize\_date\_1, fertilize\_date\_2, fertilize\_date\_3, bloom\_date, treatment\_date\_1, treatment\_type\_1, treatment\_date\_2, treatment\_type\_2, spent\_date, prune\_date

**POINT PLAN**

|  |  |  |
| --- | --- | --- |
| **Method** | **✔** | **Pts.** |
| ***Minimum Requirements*** |  |  |
| Full CRUD |  | --- |
| Foreign key & join statement |  | --- |
| MVP Plan passed off with mentor |  | --- |
| ***Core (maxes at 45)*** |  | **(40-45)** |
| Media Queries – 3+ views responsive on 2 or more screen sizes |  | 15 |
| Redux – read/write to store from 2+ reducers (read only – 5pts) |  | 5-15 |
| Hooks – implemented on at least 5 components |  | 10 |
| Authentication – functional for login & registration |  | 10 |
| ***Additional Technologies*** |  | **(30)** |
| Twilio – send reminder messages through SMS |  | 10 |
| React Big Calendar – dynamically draws/displays information from database |  | 10 |
| Sass – 50% of project styling (includes variables, mix-ins, nesting/inheritance) |  | 10 |
| ***Hosting*** |  | **(15)** |
| Successfully hosted |  | 10 |
| Registered under unique domain name |  | 5 |
| ***Presentation*** |  | **(10)** |
| Purpose, MVP, technologies, no discussion of broken features/experience, 3:00 |  | 10 |
| **TOTAL** |  | **90-100** |