

# Griffith Buck Rose Website

Client: Reiman Gardens, Lindsey Smith

Advisor Mai Zheng

sdmay24-41

By Patrick Origer, Logan Schmit, Amy Hartjen, Alex Reynolds,  
Erik Sandberg, Greg Carter and Devin Amdahl

# Introduction

**Project:** Informative website on the unique roses created by Dr. Griffith Buck which are stored at Reiman Gardens

## Objective:

- Informative: Basic rose info, Griffith Buck and Reiman Gardens
- Maintainable: The website shall allow Reiman Garden employees to edit the website's information via simple forms
- Mobile: The website needs to be structured for mobile applications
- Overall: Create a website to make the general public more aware and informed of the Griffith Buck roses in a highly readable way



<http://griffithbuck.ece.iastate.edu:3000/Home>

# Users

## Who:

- Rose cultivators and Buck Rose enthusiasts
- Elderly
- Average user needs highly intuitive and user friendly website

## Uses:

- Information gathering for specific roses
- Filter/search
- Plant Care
  - Rose bloom
  - Pest Appearance

## Why:

- Dr. Griffith Bucks famous research into plant hardiness and disease-resistance
- Over 60 industry sources including nurseries in the US, Canada, England, New Zealand, Denmark, Norway, and India

## Context:

- Global Impact: The public will be able to learn about the roses as well as get help with cultivation and maintenance
- Economic Impact: Marketing for Reiman Gardens and Buck Rose vendors

# Project Overview

## Project Management:

- Github
- Discord

## Design Tools:

- Figma
- LucidChart
- Draw.io

## Frontend Stack:

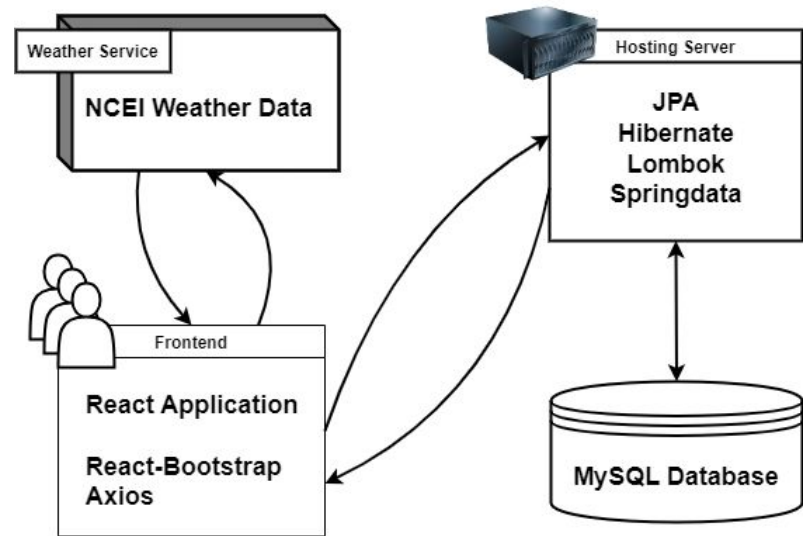
- React.js
- Axios
- Bootstrap

## Database:

- MySQL
- Flower Information
- Page Information
- Vendor statuses

## Backend Stack:

- Docker
- MySQL
- JPA
- Spring/Boot
- Hibernate
- Lombok
- Spring Data



# Major Components

**Homepage**

**Individual Rose Pages**

**Mobile Version**

**Admin Pages**

**Degree Day Calculator**

# Homepage

## Fluid Key

- Dynamically filters flowers based on any combination of there dozens of characteristics.
- Integrates with pagination
- Inclusive or exclusive filtering

## Fact/Rose of the Day

- Based on local time zone
- Pulls random flower/fact and display on carousel

## Info Pages

- About Us, Griffith Buck, Glossary
- Dynamic Display
- Navigation of different section via banner buttons

# Individual Rose Pages

## General Information:

- Cultivation year, color, name, description, height, parentage, descendants, location, hardiness, notes, and disease resistance

## Bloom Tracker:

- Shows week that individual flowers bloom

## Vendors:

- Includes vendor pricing, last updated, name and website link
- Dynamically displays any amount of vendors

## Dynamic Routing:

- Unique dynamically created link for each flower
- More shareable

# Admin Page

## Login:

- Admin pages hidden from general user
- Admin users can log in with username & password

## Navigation:

- Landing page lists all tables
- Information about each table on the admin home page

## Updating:

- Admin page loads specific database table
- Admin user can select row they want, then they can edit the information in dynamically made forms
- Admin User can send edited row as an update, addition or deletion into that table



# Degree Day Calculator

## Quick Overview:

- Tracking when pests become threats to Buck Roses
- Calculates based on weather data from stations across Iowa
- Calculation methods: Average, Modified Average/Custom, Sine

$$\frac{1}{\pi} \left\{ \left( \frac{T_{\max} + T_{\min}}{2} - T_L \right) (\theta_2 - \theta_1) + \alpha [\cos(\theta_1) - \cos(\theta_2)] + (T_U - T_L) \left( \frac{\pi}{2} - \theta_2 \right) \right\}$$

$$\theta_1 = \sin^{-1} \left[ \left( T_L - \frac{T_{\max} + T_{\min}}{2} \right) \div \alpha \right]$$

$$\theta_2 = \sin^{-1} \left[ \left( T_U - \frac{T_{\max} + T_{\min}}{2} \right) \div \alpha \right]$$

## Input/Output Briefing:

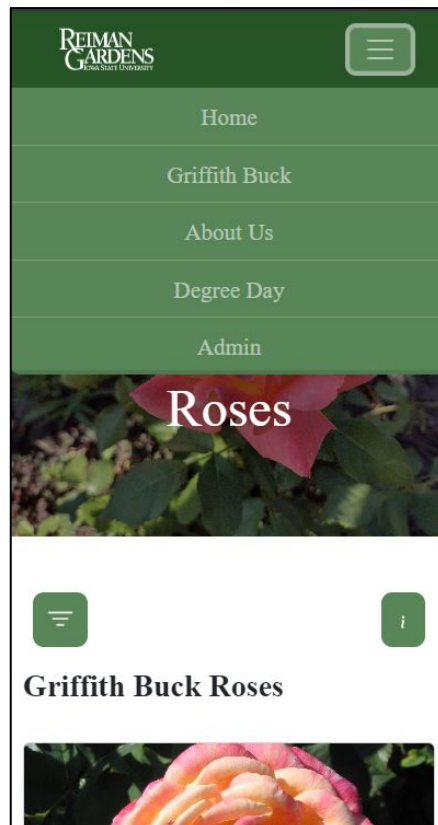
- Inputs: Station, Date Range, Daily Temps, Temp Bounds
- Output table:

Day	Date	Min Temp	Max Temp	Degree Days	Cumul. Degree Days
51	2024-02-20	25	64	7	35.5
52	2024-02-21	33	64	7	42.5
53	2024-02-22	36	65	7.5	50
54	2024-02-23	28	54	2	52
55	2024-02-24	17	51	0.5	52.5
56	2024-02-25	25	67	8.5	61
57	2024-02-26	29	78	14	75
58	2024-02-27	14	61	5.5	80.5
59	2024-02-28	7	31	0	80.5
60	2024-02-29	14	58	4	84.5
61	2024-03-01	24	56	3	87.5
62	2024-03-02	28	69	9.5	97
63	2024-03-03	37	79	14.5	111.5
64	2024-03-04	31	46	0	111.5
65	2024-03-05	25	58	4	115.5
66	2024-03-06	22	59	4.5	120
67	2024-03-07	35	56	3	123
68	2024-03-08	30	46	0	123

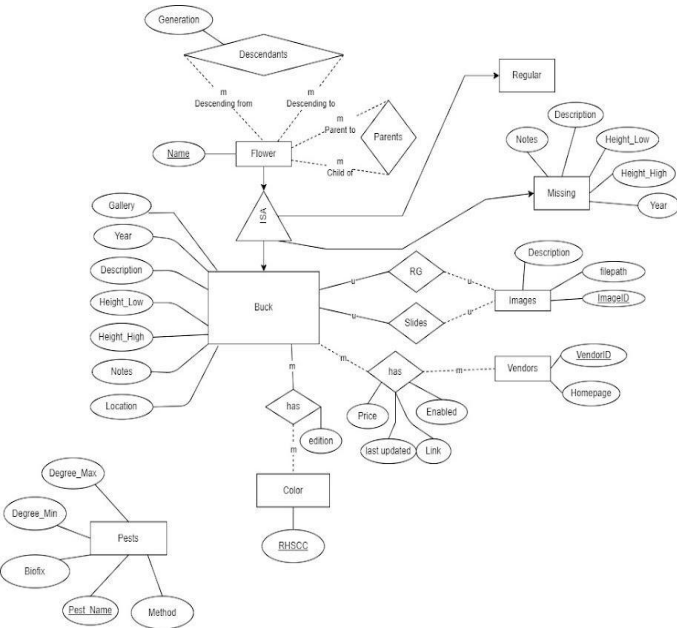
# Mobile Version

## Mobile Version Implementation:

- CSS media queries
- Bootstrap
  - Breakpoints, grid system, etc.
- Dynamic sizing
- Separate implementation for navbar and footer
- Horizontal scrolling for some features



# Database Design

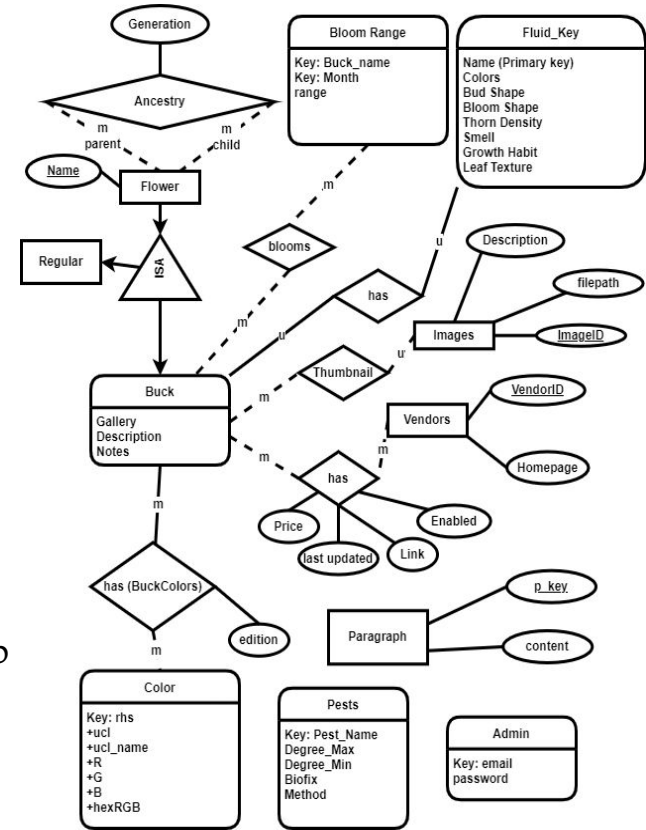


## Design Choices

- MySQL
- Feedback: Graph Database

## Design Changes

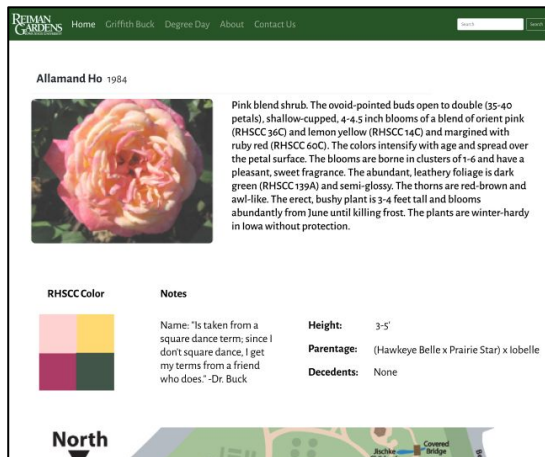
- Remove 'missing' flowers table
- Bloom days table addition
- Admin authentication table
- Fluid Key attributes expanded
- Paragraph information in database vs hardcoded
- Change thumbnails relationship
- Merge parents & descendants table into 'Ancestry'



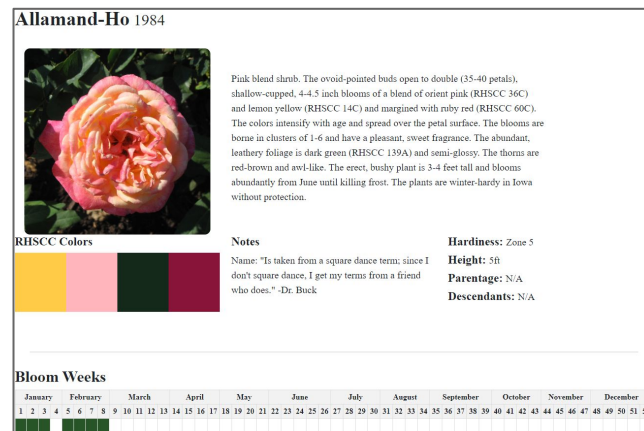
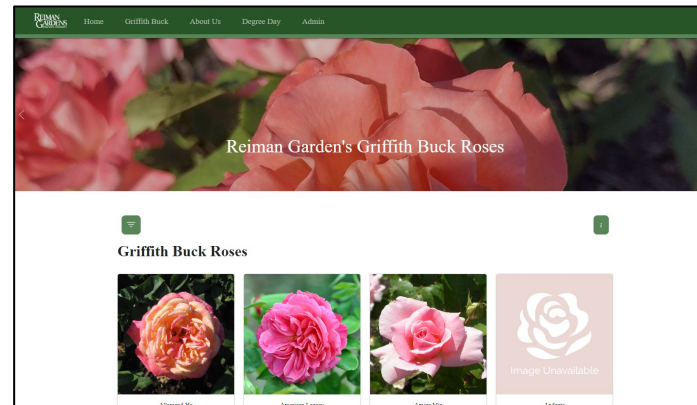
# Frontend Design



- Fluid Key is collapsible
- Footer has contact us information
- Glossary Page
- Removed global search feature



- Removed dynamic map
- Bloom days separate from degree days



# Challenges

## Data:

- Frequent database/schema changes
- Data retrieval
  - Flower photos in the fall
  - Reiman Gardens map
  - Disease resistance
  - Bloom ranges
  - Fluid Key attributes
  - Pests

## Security:

- Originally using Okta
- ITS switching from Okta to Azure
- Implementing authentication and session management ourselves

## Other:

- Degree Day/weather API
- Server storage of images
- Dynamic vendor information
- Larger group
  - Double-edged sword

# Testing

## Libraries:

- JUnit
- Serenity
- Mockito
- React's Testing Library

## Types:

- Unit
- System
- Acceptance
- Beta Testing since April 7th

## Process:

- Tests run in Build and Pipeline
- Manual testing to ensure functionality
- Reviews and Approvals before merges

```
at console.<anonymous> (node_modules/jest-mock/build/index.js:794:25)
(node:17112) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/Main.test.js (6.098 s)

Test Suites: 8 passed, 8 total
Tests:      31 passed, 31 total
Snapshots:  0 total
```

```
(node:6924) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/FlowerCard.test.js
(node:7604) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/Carousel.test.js
(node:19260) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/Glossary.test.js
(node:20672) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/About.test.js (5.114 s)
(node:15936) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/Buck.test.js (5.16 s)
PASS src/Tests/Hello.test.js
(node:17420) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use
a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/Tests/Login.test.js (5.833 s)
● Console
```

# Conclusions

## Progress:

- We were able to complete the original goal of building an easily maintainable website for Reiman Gardens
- Management of website requires little to no code knowledge
- Mobile design is fully implemented
- Advanced degree day calculator to replace Michigan State University's
- Information displayed can help cultivators by given them pesticide times and bloom ranges

## Value:

- Provides open, easy access to information about Griffith Buck Roses and Reiman Gardens

## Future Client Goal:

- CSV Upload
- Better Image Storage Solution
- Dynamic Linking
- Cloud Architecture

**Questions?**



# Thank You!

<https://sdmay24-41.sd.ece.iastate.edu>

# Who are We?

Amy Hartjen - Client Interaction, Organization, Frontend Design

Erik Sandberg - Full-Stack, Database Administrator

Devin Amdahl - DevOps, Scrum Master, Full-Stack

Patrick Origer - Research, Third Party Manager, Degree Days

Alex Reynolds - Frontend Lead

Greg Carter - Full-Stack, Connections, Backend

Logan Schmit - Backend Helper, Lead Testing