

SE/ComS319 Construction to User Interfaces

Fall 2023

Isaac Lo - isaaclo@iastate.edu

Siddhartha Gudipudi - siddu123@iastate.edu

Dr. Abraham Aldaco - aaldaco@iastate.edu

Dec 13, 2023

# Final Project Documentation

*Vesper Flora*

# Index

- [Project description](#)
- [Software functionality diagram](#)
- [Files and directory architecture](#)
  - [Description and diagram of files and directory architecture](#)
  - [Description of client – server architecture](#)
  - [Database and API diagrams](#)
  - [Description and diagram of the logical architecture](#)
  - [Explanation of each web view:](#)
- [Installation manual:](#)
  - [Prerequisites](#)
  - [Step-by-Step Installation Guide](#)
- [Code Documentation:](#)
  - [Backend Code Documentation:](#)
  - [Frontend Code Documentation](#)
  - [Verbatim Code:](#)
    - [db.js:](#)
    - [Index.js](#)
    - [appbackend.js](#)
  - [appfrontend.js](#)
- [Final Comments](#)

# Project description

## Introduction:

The flora community grows ever-vibrant as the world becomes increasingly digital and individualistic. Our final project, *Vesper Flora*, aims to capture this appeal for the naturally beautiful and vibrant and join it with the self-investment aspects of health and do-it-yourself guides. *Vesper Flora* will be a resource that allows personally passionate users to connect with an excited community of like-minded individuals and be the exchange place of flora knowledge.

The naturality and the simple beauty of plant life have become a thoughtful reclusion for the busy and anxious modern citizen. Our project, *Vesper Flora*, will be the guide to navigate the flora knowledge landscape and a supplier for passionate and authentic products.

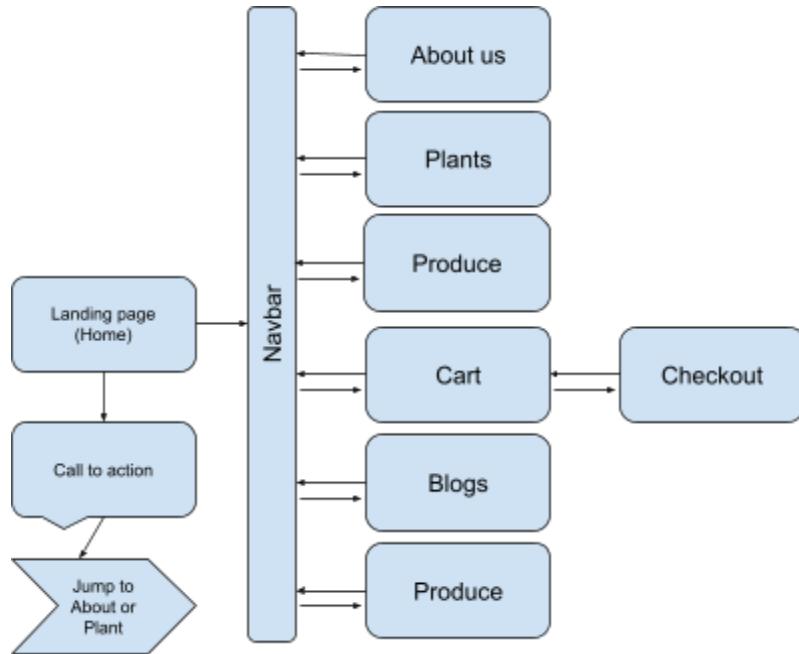
Our initiative aims to apply our web development and e-commerce concepts in a creative, elegant, and educational setting. Due to their widespread appeal, plant organics make this exercise an engaging topic to carry our development skills into a project we're eager to create.

## Objectives:

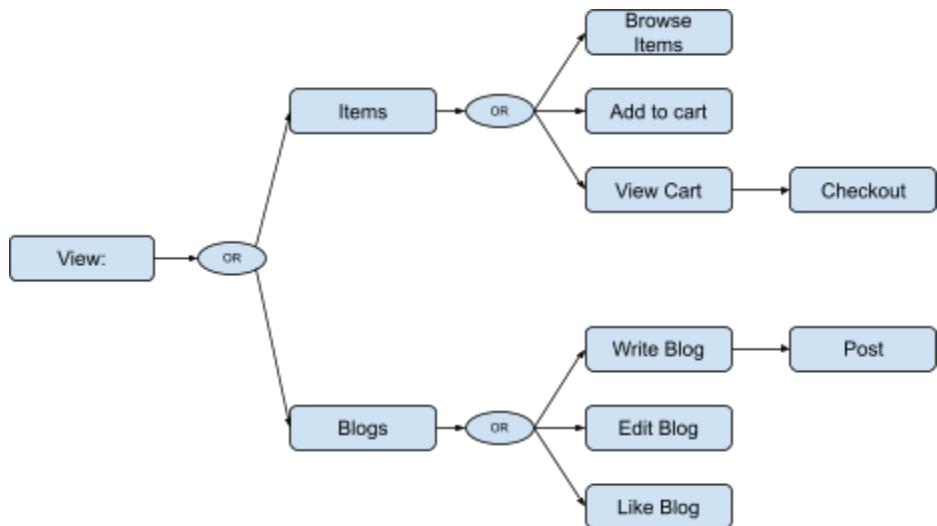
Our final project's main goal is to create a powerful e-commerce and online guide website and to obtain practical experience in web design and development along the way. We aim to achieve elegance in its design and strength in its functionality. We will address the problems of expanding cheap and unhealthy plant products among popular resellers, owners' care/skill ignorance, and lacking a trusted, centralized location for plant enthusiasts to gather. **Vesper** will solve this problem by creating a dependable source for excellent plants, expert and easy guides, and an electric flora-forum community for any nature lover worldwide.

# Software functionality diagram

Navigation Diagram:

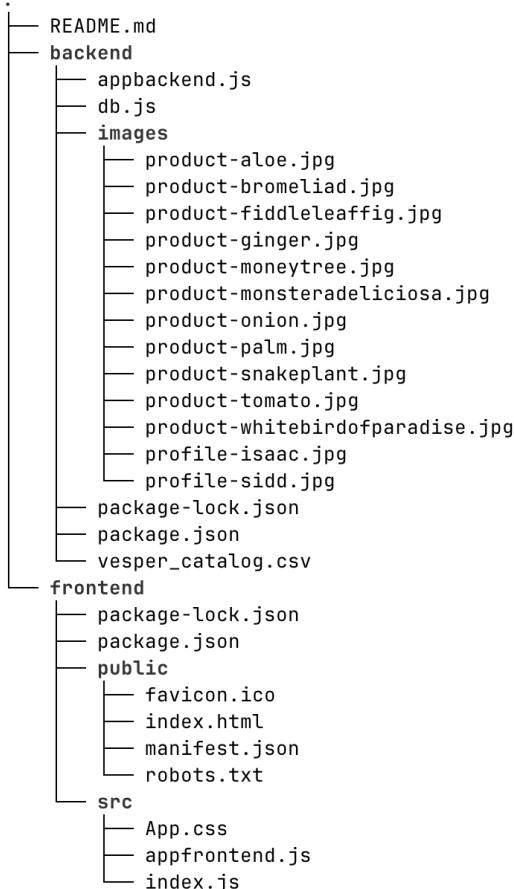


Functionality Diagram:



# Files and directory architecture

## Description and diagram of files and directory architecture

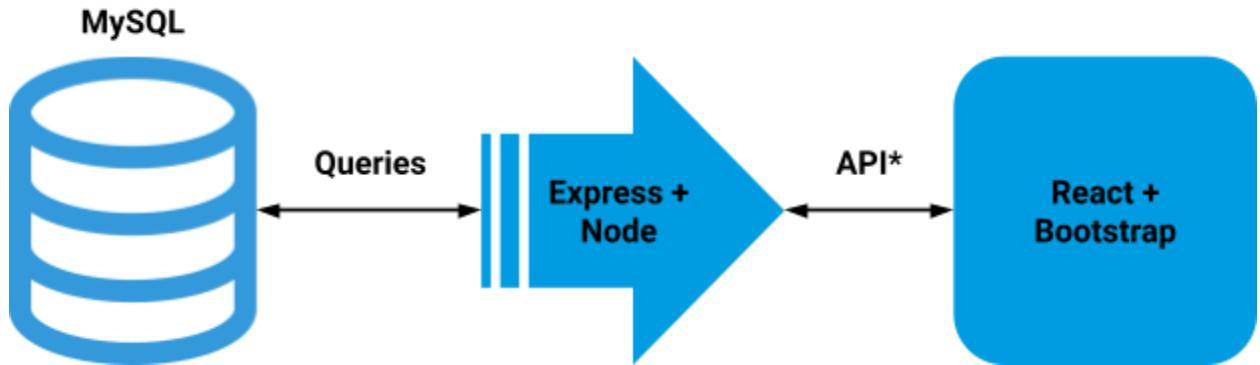


The project is separated into the backend and frontend folders.

**Backend:** The backend Express application logic is found in `appbackend.js`. The connection to the database (in this case MySQL) is handled by the `db.js` file. The `images` folder is a statically hosted (on Express) folder that makes the images accessible for the frontend. The contents of this folder are not necessarily the ones shown. The `package-lock.json` and the `package.json` hold the dependencies and version information for the backend side of the project. The final file `vesper_catalog.csv` is a file containing some example data for the MySQL database for easy access.

**Frontend:** The frontend React application logic is found in `appfrontend.js`. All non-Bootstrap is found in `App.css`. `index.js` loads the main app from `appfrontend.js`. The `package-lock.json` and the `package.json` hold the dependencies and version information for the frontend side of the project. The `favicon.ico` file is the thumbnail for web browsers. The `index.html` page is the main page where the root of the app is loaded. The `manifest.json` file holds more information about the frontend app. The `robots.txt` file tells automated web bots to ignore all the pages in this project when scraping the internet.

## Description of client – server architecture



\* API described in next section

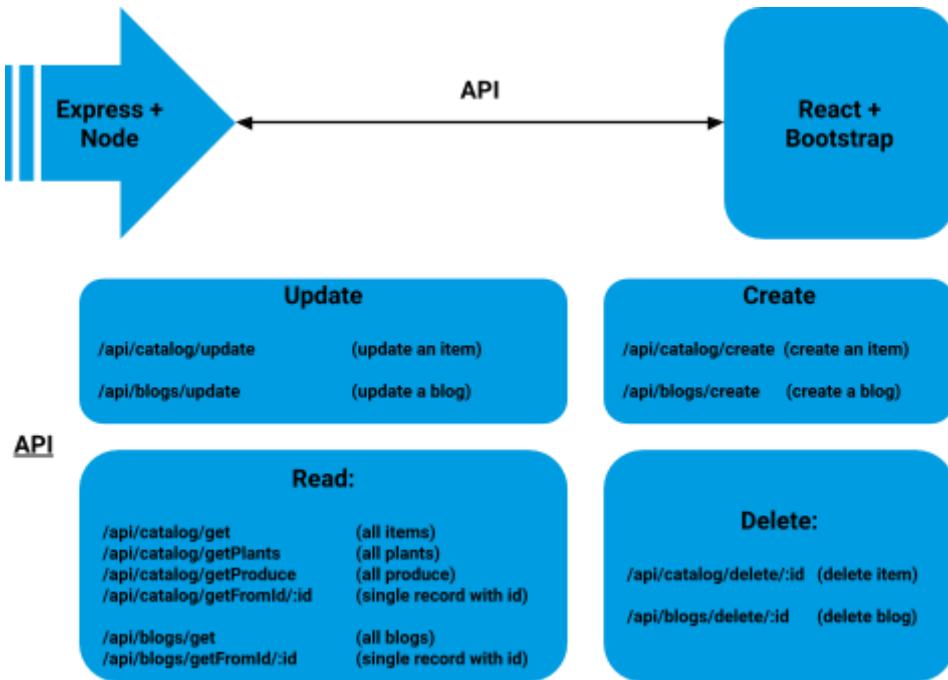
The *Vesper Flora* project is a SERN application.

The frontend uses [React](#). This allows the project frontend to have state and to render intelligently and efficiently. The frontend also uses [Bootstrap](#) to style the webpage. The project uses a [Node.js](#) backend. Node hosts the backend using [Express](#). Express creates the API endpoints used for the CRUD operations (Create, Read, Update, Delete) related to the blogs and items Vesper Flora host. The database used is a [MySQL](#) database. Express interacts with the database by using the [mysql2](#) node package.

Dataflows in this way: The MySQL database holds all records related to the project. The React frontend requests information to the Node.js backend through the Express API endpoints. The Node.js backend then uses express to query the MySQL database. Express receives a result from the database, Node sends this response back to the frontend through a JSON body.

Express is a critical part of the dataflow. Express abstracts the database into API endpoint calls—specifically fetch calls in the frontend.

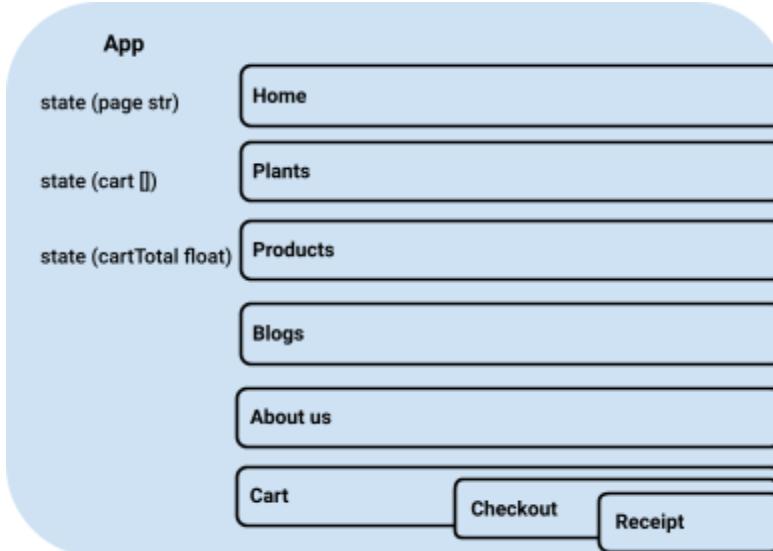
## Database and API diagrams



API includes all CRUD operations. Dataflow explained in the previous [section](#). There are separate API calls for the blog and the catalog tables. For an example of the code used to create the api endpoints see the [code](#) section or the [previous](#) section.

## Description and diagram of the logical architecture

React Module Diagram:



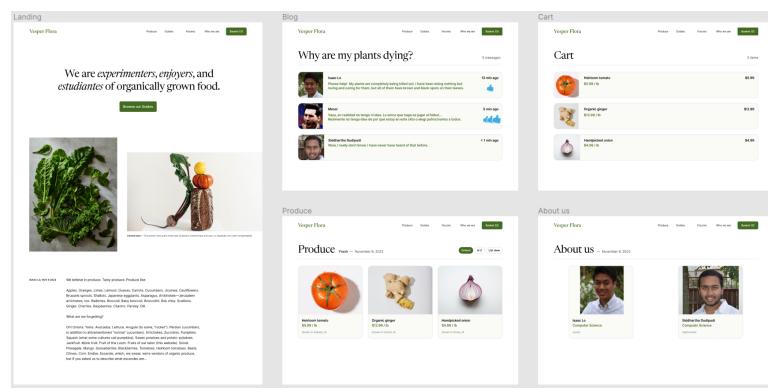
The *Vesper Flora* project uses a main component called App and a single state, called “page,” to control the main content of the webpage. This state is simply a string. The main App component uses a check for the page string and accordingly renders the correct component.

For more info on view flow: see [software functionality section](#), navigation diagram and functionality diagram, for more information about the view logic.

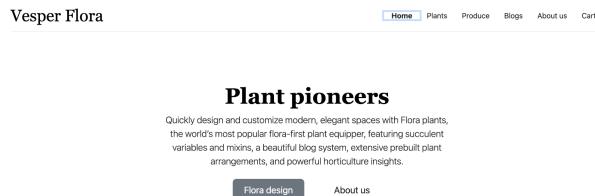
Views Figma:

Link to initial Figma designs: [Vesper Flora](#)

Image of designs:



Explanation of each web view:



Main Landing Page.

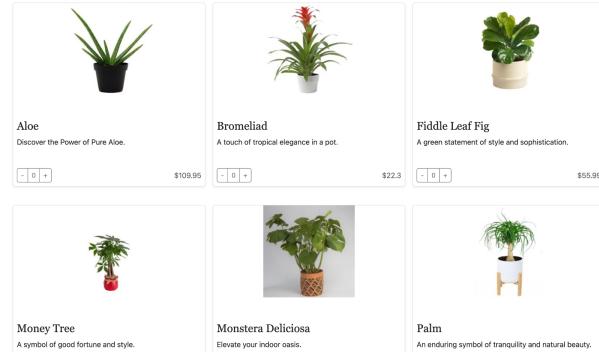
Navbar for navigation.

Main call to action “hero.”

Two buttons. One navigates to plants catalog. The other to the about page.

**Design**  
**Plants**

Catalog | Dec 3

**Plant Catalog Page.**

Navbar for navigation.

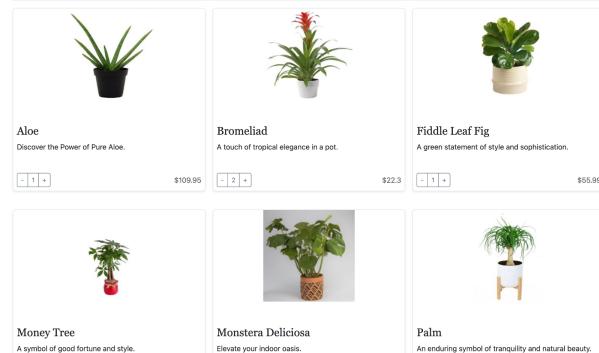
Explanation header ("Plants").

Plant catalog dynamically build from MySQL database.

Add or subtract plants in cart with + or - buttons. Seen below:

**Design**  
**Plants**

Catalog | Dec 3



Cart in navbar changes color, and rerenders dynamically with the number of items in the cart.

## Produce

## Fresh produce

Catalog | Dec 3



Produce Catalog Page.

Navbar for navigation.

Explanation header ("Fresh produce").

Plant catalog dynamically build from MySQL database.

Add or subtract produce in cart with + or - buttons. Seen below:

## Produce

## Fresh produce

Catalog | Dec 3



Cart in navbar changes color, and rerenders dynamically with the number of items in the cart.

## Learn

## Prompt of the day: How do you care for orchids?

Blogs | Dec 12

## "You shouldn't overwater"

17

Isaac Lo

"The number one thing you shouldn't do is overwater your orchid! Proper watering is crucial for the health and well-being of orchids, and avoiding overwatering is a key practice that can lead to several benefits. Orchids, known for their exotic and delicate appearance, are adapted to thrive in specific conditions. Overwatering can lead to root rot, a common issue for orchids, as it deprives the roots of essential oxygen. By allowing the orchid's growing medium to dry out between waterings, excess moisture is prevented, promoting aeration and maintaining a healthy balance. This practice not only helps prevent root diseases but also encourages the development of a robust root system. Additionally, avoiding overwatering supports the orchid's natural growth cycle, mimicking the intermittent periods of wet and dry conditions they experience in their native habitats. Overall, proper watering habits contribute to the longevity and vibrancy of orchids, ensuring they flourish and produce stunning blooms in a well-balanced environment."

## "Overwatering for dayzz"

3

CrazyPerson655

"Not overwatering orchids is essential for maintaining their health and longevity. Overwatering can lead to root rot, depriving the roots of vital oxygen. Allowing the growing medium to dry out between waterings prevents excess moisture, fostering aeration and supporting a robust root system. This practice not only prevents root diseases but also aligns with the orchid's natural growth cycle, contributing to vibrant blooms and overall well-being."

## Bruh

0

BingusSkeg

Ilo dolores autem placeat. Fugiat commodi autem voluptas. Sepiente rem dolorem voluptates doloribus itaque quis ullam. Reprehenderit molestias natus id autem occaecati est veniam. Maxime mollitia est beatiae autem. NIHIL et et doloremque. Libero excepturi ab reprehenderit et consequatur saepe sed. Voluptates tempora quia similique reiciendis qui omnis. ipsa minima cumque nihil at non aut. Quos lure non sit quia aut enim. Vitiae cumque eum ullam. Voluptates debitis ratione

## Blog Page.

Navbar for navigation.

Prompt of the day header ("How do you care for orchids?")

Blog posts dynamically build from MySQL database.

Like or dislike blog posts  or  buttons. Seen below:

## Learn

## Prompt of the day: How do you care for orchids?

Blogs | Dec 12

## "You shouldn't overwater"

18

Isaac Lo

"The number one thing you shouldn't do is overwater your orchid! Proper watering is crucial for the health and well-being of orchids, and avoiding overwatering is a key practice that can lead to several benefits. Orchids, known for their exotic and delicate appearance, are adapted to thrive in specific conditions. Overwatering can lead to root rot, a common issue for orchids, as it deprives the roots of essential oxygen. By allowing the orchid's growing medium to dry out between waterings, excess moisture is prevented, promoting aeration and maintaining a healthy balance. This practice not only helps prevent root diseases but also encourages the development of a robust root system. Additionally, avoiding overwatering supports the orchid's natural growth cycle, mimicking the intermittent periods of wet and dry conditions they experience in their native habitats. Overall, proper watering habits contribute to the longevity and vibrancy of orchids, ensuring they flourish and produce stunning blooms in a well-balanced environment."

## "Overwatering for dayzz"

2

CrazyPerson655

"Not overwatering orchids is essential for maintaining their health and longevity. Overwatering can lead to root rot, depriving the roots of vital oxygen. Allowing the growing medium to dry out between waterings prevents excess moisture, fostering aeration and supporting a robust root system. This practice not only prevents root diseases but also aligns with the orchid's natural growth cycle, contributing to vibrant blooms and overall well-being."

## Bruh

0

BingusSkeg

Ilo dolores autem placeat. Fugiat commodi autem voluptas. Sepiente rem dolorem voluptates doloribus itaque quis ullam. Reprehenderit molestias natus id autem occaecati est veniam. Maxime mollitia est beatiae autem. NIHIL et et doloremque. Libero excepturi ab reprehenderit et consequatur saepe sed. Voluptates tempora quia similique reiciendis qui omnis. ipsa minima cumque nihil at non aut. Quos lure non sit quia aut enim. Vitiae cumque eum ullam. Voluptates debitis ratione

Likes and disliked allotted to posts. Only allows for one like or dislike per page visit (lack of user logging functionality in beta).

a key practice that can lead to several benefits. Orchids, known for their exotic and delicate appearance, are adapted to thrive in specific conditions. Overwatering can lead to root rot, a common issue for orchids, as it deprives the roots of essential oxygen. By allowing the orchid's growing medium to dry out between waterings, you maintain a healthy moisture balance. This practice not only prevents root diseases but also aligns with the orchid's natural growth cycle, contributing to vibrant blooms and overall well-being.

**"Overwatering for dayzz"**

CrazyPerson555

"Not overwatering orchids is essential for maintaining their health and longevity. Overwatering can lead to root rot, depriving the roots of vital oxygen. Allowing the growing medium to dry out between waterings prevents excess moisture, fostering aeration and supporting a robust root system. This practice not only prevents root diseases but also aligns with the orchid's natural growth cycle, contributing to vibrant blooms and overall well-being."

**Bruh**

BingusSkeg

Ilo dolores autem placeat. Fugiat commodi autem voluptas. Sapiente rem dolorem voluptates doloribus itaque quis ullam. Reprehenderit molestias natus id autem occaecati est veritatem. Maxime mollitia est beatas autem. NIH et et dolorumque. Libero exceptus ab reprehenderit et consequatur sapio sed. Voluptates tempora quia similiqure reciendis qui omnis. Ipsa minima cumque nihil at non aut. Quos iure non sit quia aut enim. Vitae cumque eum ullam. Voluptates debitis ratione laboriosam tenuerunt excepturi. Ilo et animi at soluta necessitatibus quibusdam nisi veritatis. Iusto blanditiis quod error. Sit quidem neque voluptates ad. Repellendus quos nihil maiores aliquam molestiae. Facere sunt ipsam qui et qui et. Incidunt tempore nulla dolorem suscipit voluptas accusantium officia expedita.

**asdf**

BingusSkeg

Hjello vdeo

**Blog me**

Add the bottom of the blog page is the “Blog me” button. Press brings form view below:

**Bruh**

BingusSkeg

Ilo dolores autem placeat. Fugiat commodi autem voluptas. Sapiente rem dolorem voluptates doloribus itaque quis ullam. Reprehenderit molestias natus id autem occaecati est veritatem. Maxime mollitia est beatas autem. NIH et et dolorumque. Libero exceptus ab reprehenderit et consequatur sapio sed. Voluptates tempora quia similiqure reciendis qui omnis. Ipsa minima cumque nihil at non aut. Quos iure non sit quia aut enim. Vitae cumque eum ullam. Voluptates debitis ratione laboriosam tenuerunt excepturi. Ilo et animi at soluta necessitatibus quibusdam nisi veritatis. Iusto blanditiis quod error. Sit quidem neque voluptates ad. Repellendus quos nihil maiores aliquam molestiae. Facere sunt ipsam qui et qui et. Incidunt tempore nulla dolorem suscipit voluptas accusantium officia expedita.

**asdf**

BingusSkeg

Hjello vdeo

**Title**

Your blog post text here...

Adding information and submitting brings sends the blog post to be a record in the database.

**"Overwatering for dayzz"**

CrazyPerson555

"Not overwatering orchids is essential for maintaining their health and longevity. Overwatering can lead to root rot, depriving the roots of vital oxygen. Allowing the growing medium to dry out between waterings prevents excess moisture, fostering aeration and supporting a robust root system. This practice not only prevents root diseases but also aligns with the orchid's natural growth cycle, contributing to vibrant blooms and overall well-being."

**Bruh**

BingusSkeg

Ilo dolores autem placeat. Fugiat commodi autem voluptas. Sapiente rem dolorem voluptates doloribus itaque quis ullam. Reprehenderit molestias natus id autem occaecati est veritatem. Maxime mollitia est beatas autem. NIH et et dolorumque. Libero exceptus ab reprehenderit et consequatur sapio sed. Voluptates tempora quia similiqure reciendis qui omnis. Ipsa minima cumque nihil at non aut. Quos iure non sit quia aut enim. Vitae cumque eum ullam. Voluptates debitis ratione laboriosam tenuerunt excepturi. Ilo et animi at soluta necessitatibus quibusdam nisi veritatis. Iusto blanditiis quod error. Sit quidem neque voluptates ad. Repellendus quos nihil maiores aliquam molestiae. Facere sunt ipsam qui et qui et. Incidunt tempore nulla dolorem suscipit voluptas accusantium officia expedita.

**asdf**

BingusSkeg

Hjello vdeo

**Documentation**

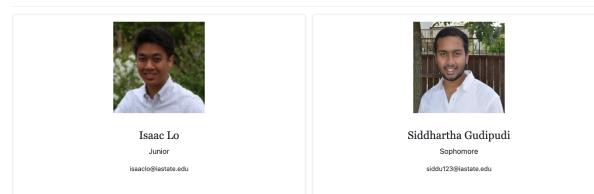
Example Author

Hello Documentation! Nice to see you. Hope you're having a wonderful day! Thanks, Example Author.

**Blog me**

About us  
**Two computer scientists**

3 Dec 2023  
 SEComS319 Construction of User Interfaces, Fall 2023  
 Dr. Abraham N. Aldaco Gastelum - [aldaco@isastate.edu](mailto:aldaco@isastate.edu)



## About us Page.

Navbar for navigation.

Attention-grabbing explanation header.

Images pulled from database (Express static hosting).

Vesper Flora

Home Plants Produce Blogs About us Cart (10)

**Payment Information**

First name	Last name	
you@example.com		
Address	1234 Main St	
Country	State	Zip
Choose...	Choose...	
Credit card number	XXXX-XXXX-XXXX-XXXX	
Expiration Date (MM/YY)	MM/YY	
CVV (3-digit)	CVV (3-digit)	

**Your cart** 10

Item	Quantity	Price
Heirloom Tomato	x1	\$10.99
Handpicked Onion	x1	\$0.99
Organic Ginger	x1	\$15.99
Buddle Leaf Fig	x1	\$55.99
Aloe	x3	\$329.85
White Bird of Paradise	x1	\$10.99
Monstera Deliciosa	x2	\$33.6
Tax (%)		\$32.72
Total (USD)		<b>\$600.12</b>

**Place Order**

## Cart Page

Navbar for navigation.

Order form on left. Cart receipt on right.

Vesper Flora

Home Plants Produce Blogs About us Cart (10)

**Payment Information**

First name	Last name	
Isaac	Lo	
Email	isaaclo@instate.edu	
Address	1234 Main street	
Country	State	Zip
United States	Iowa	50010
Credit card number 1234-5678-9123-4567		
Expiration Date (MM/YY) 12/12		
CVV (3-digit) 123		

**Your cart** (10)

Heirloom Tomato	\$10.99
Handpicked Onion	\$9.99
Organic Ginger	\$15.99
Fiddle Leaf Fig	\$55.99
Aloe	\$329.85
White Bird of Paradise	\$10.99
Monstera Deliciosa	\$33.6
Tax (%)	\$32.72
Total (USD)	<b>\$500.12</b>

**Place Order**

Cart Page with form valid information entered. Place order below:

Vesper Flora

Home Plants Produce Blogs About us Cart

**Flora Receipt:**

**Your cart** (10)

Heirloom Tomato	\$10.99
Handpicked Onion	\$9.99
Organic Ginger	\$15.99
Fiddle Leaf Fig	\$55.99
Aloe	\$329.85
White Bird of Paradise	\$10.99
Monstera Deliciosa	\$33.6
Tax (%)	\$32.72
Total (USD)	<b>\$500.12</b>

Thank you for your purchase.

Isaac Lo  
isaaclo@instate.edu  
1234 Main street  
United States, Iowa 50010

With invalid information entered:

Please enter your first name.	Please enter your last name.	
Email: you@example.com	Please enter a valid email address.	
Address: 1234 Main St	Please enter your address.	
Country: Choose...	State: Choose...	Zip: Please choose your country. Please choose your state. Please enter a valid zip code.
Credit card number: XXXX-XXXX-XXXX-XXXX		
Please enter a valid credit card number.		
Expiration Date (MM/YY) MM/YY		
Please enter a valid expiration date (MM/YY).		
CVV (3-digit) CVV (3-digit)		
Please enter a valid 3-digit CVV number.		

# Installation manual:

Follow these steps to successfully install and run *Vesper Flora* on your local environment.

## Prerequisites

Before starting the installation process, make sure you have the following software installed on your system (A good guide is also found [here](#) on our GitHub repo):

- Node.js, npm (Node Package Manager), MySQL

## Step-by-Step Installation Guide

1. Make sure to download the zip of the application.

2. Navigate to Project Directory:

Change into the project directory: cd *vesper-flora*

### 3a. For BACKEND:

-> Install Backend Dependencies:

- Navigate back to the main project directory: cd ..
- Move into the frontend folder: cd backend
- Install frontend dependencies using npm: npm install
- You will need to have MySQL server and “\$ sql.server start” it.

### 3b. For FRONTEND:

-> Install Frontend Dependencies:

- Navigate back to the main project directory: cd ..
- Move into the frontend folder: cd frontend
- Install frontend dependencies using npm: npm install

### 4. Start the Application:

- Navigate back to the main project directory: cd ..

- Start the backend server: cd backend

node appbackend.js

- Open a new terminal window and start the frontend development server:

:cd frontend

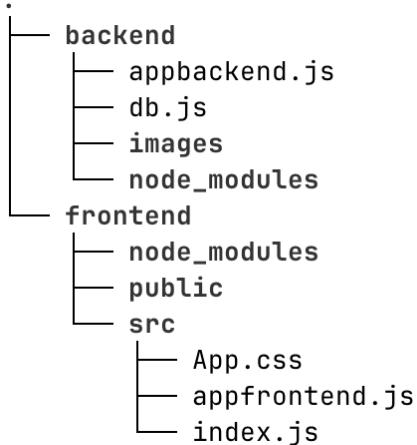
Npm start

5. Access the Application: Open your web browser and go to `http://localhost:3000` to access the *Vesper Flora* application.

=> Congratulations! You have successfully installed Vesper Flora on your local environment. Explore the website, connect with the flora community, and enjoy the experience.

# Code Documentation:

Basic File Structure of the project (full explanation [here](#)):



## Backend Code Documentation:

### appbackend.js

This file contains the main logic for the Express.js application, serving as the backend for *Vesper Flora*.

#### - Endpoints:

- /api/blogs: Handles CRUD operations for blog-related data.
- /api/catalog: Manages CRUD operations for catalog items.

#### - Middleware:

- express.json(): Parses incoming requests with JSON payloads.
- cors: Handles Cross-Origin Resource Sharing.
- morgan: HTTP request logger middleware.

#### - Database Connection:

- db.js: Manages the connection to the MySQL database using the `mysql2` package.

### db.js

Responsible for establishing and managing the connection to the MySQL database.

#### - Functions:

- connectToDatabase(): Connects to the MySQL database.
- queryDatabase(query): Executes a SQL query on the connected database.

# Frontend Code Documentation

## appfrontend.js

This file contains the main logic for the React application, serving as the frontend for *Vesper Flora*.

### - Components:

- App: The main component managing the state and rendering other components based on the current state.
- Other components: Additional components such as `BlogList`, `CatalogItems`, etc.

### - State:

- page: Controls the main content of the webpage by rendering different components based on its value.

### - API Requests:

- Utilizes `fetch` to make requests to the Express.js API endpoints for fetching and manipulating data.

## App.css

This file contains the styling for the React components, excluding Bootstrap styles.

## index.js

Loads the main `App` component into the root element of the HTML document.

## Database Schema Documentation

## vesper\_catalog.sql

Defines the schema for the MySQL database used by *Vesper Flora*, including tables for blogs and catalog items.

## Additional Information

### - Dependencies:

- `package.json` and `package-lock.json` in both the frontend and backend folders contain information about project dependencies and their versions.

### - Static Files:

- Images in the `backend/images` folder are statically hosted and accessible for the frontend.

## Verbatim Code:

### db.js:

```
const mysql = require('mysql2')

const db = mysql.createConnection({
  host: "127.0.0.1",
  user: "isaaclo",
  password: "pooPooP33P33",
  database: "secoms319"
})

module.exports = db;
```

### index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './appfrontend.js';
import 'bootstrap/dist/css/bootstrap.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

### appbackend.js

```
const express = require("express");
const db = require("./db.js");
const cors = require("cors");

const app = express();

const PORT = 4000;
app.use(cors());
app.use(express.json());

app.use(express.static("public"));
app.use("/images", express.static("images"));
```

```

app.listen(PORT, () => {
    console.log(`Server is running on ${PORT}`);
})

// Route to get all items in catalog
app.get("/api/catalog/get", (_, res) => {
    db.query("SELECT * FROM vesper_catalog", (err, result) => {
        if (err) {
            console.log(err);
        }
        res.send(result);
    });
})

// Route to get all plants
app.get("/api/catalog/getPlants", (_, res) => {
    db.query("SELECT * FROM vesper_catalog WHERE category = 'plants'", (err, result)
=> {
        if (err) {
            console.log(err);
        }
        res.send(result);
    });
})

// Route to get all produce
app.get("/api/catalog/getProduce", (_, res) => {
    db.query("SELECT * FROM vesper_catalog WHERE category = 'produce'", (err, result)
=> {
        if (err) {
            console.log(err);
        }
        res.send(result);
    });
})

// Route to get one record
app.get("/api/catalog/getFromId/:id", (req, res) => {
    const id = req.params.id;
    db.query(
        "SELECT * FROM vesper_catalog WHERE id = ?", id,
        (err, result) => {
            if (err) {
                console.log(err);
            }
        }
    );
})

```

```

        res.send(result);
    }
);
});

// Route for creating a record
app.post("/api/catalog/create", (req, res) => {
    const id = req.body.id;
    const title = req.body.title;
    const price = req.body.price;
    const description = req.body.description;
    const category = req.body.category;
    const image = req.body.image;
    const rating = req.body.rating;
    console.log(id, title, price, description, category, image, rating);
    db.query(
        "INSERT INTO vesper_catalog (id, title, price, description, category, image, rating) VALUES (?,?,?,?,?,?,?,?)",
        [id, title, price, description, category, image, rating],
        (err, result) => {
            if (err) {
                console.log(err);
                res.status(500).send('Internal Server Error');
            } else {
                res.status(200).send('Resource Created');
            }
            console.log(result);
        }
    );
});

// Route to delete a post
app.delete("/api/catalog/delete/:id", (req, res) => {
    const id = req.params.id;
    db.query("DELETE FROM vesper_catalog WHERE id= ?", id, (err, result) => {
        if (err) {
            console.log(err);
            res.status(500).send(err);
        } else {
            res.status(200).send(result);
        }
    });
});

// Route for update a post
app.put("/api/catalog/update", (req, res) => {

```

```

const id = req.body.id;
const title = req.body.title;
const price = req.body.price;
const description = req.body.description;
const category = req.body.category;
const image = req.body.image;
const rating = req.body.rating;
console.log(id, title, price, description, category, image, rating);
db.query(
    "UPDATE vesper_catalog SET title=?, price=?, description=?, category=?,
image=?, rating=? WHERE id=?",
    [title, price, description, category, image, rating, id],
    (err, result) => {
        if (err) {
            console.log(err);
            res.status(500).send('Internal Server Error');
        } else {
            console.log(result);
            res.status(200).send('Resource Updated');
        }
    }
);
});

// Blogs Express endpoints
//

// Route to get all blogs
app.get("/api/blogs/get", (_, res) => {
    db.query("SELECT * FROM vesper_blogs", (err, result) => {
        if (err) {
            console.log(err);
        }
        res.send(result);
    });
})

// Route to get one blog record
app.get("/api/blogs/getFromId/:id", (req, res) => {
    const id = req.params.id;
    db.query(
        "SELECT * FROM vesper_blogs WHERE id = ?", id,
        (err, result) => {
            if (err) {
                console.log(err);
            }
    }
})

```

```

        res.send(result);
    }
);
});

// Route for creating a record
app.post("/api/blogs/create", (req, res) => {
    const id = req.body.id;
    const title = req.body.title;
    const author = req.body.author;
    const text = req.body.text;
    const category = req.body.category;
    const image = req.body.image;
    const likes = req.body.likes;
    console.log(id, title, category, image, likes, text);
    db.query(
        "INSERT INTO vesper_blogs (id, title, author, image, likes, text) VALUES
        (?, ?, ?, ?, ?, ?)",
        [id, title, author, image, likes, text],
        (err, result) => {
            if (err) {
                res.status(500).send(err);
                console.log(err);
            } else {
                res.status(200).send(result);
            }
            console.log(result);
        }
    );
});

// Route to delete a post
app.delete("/api/blogs/delete/", (req, res) => {
    const id = req.body.id;
    db.query("DELETE FROM vesper_blogs WHERE id= ?", id, (err, result) => {
        if (err) {
            console.log(err);
        } else {
            res.send(result);
        }
    });
});

// Route for update a post
app.put("/api/blogs/update", (req, res) => {
    const id = req.body.id;
    const title = req.body.title;

```

```

const author = req.body.author;
const text = req.body.text;
const image = req.body.image;
const likes = req.body.likes;
console.log(id, title, author, image, likes, text);
db.query(
    "UPDATE vesper_blogs SET title=?, author=?, image=?, likes=?, text=? WHERE
id=?",
    [title, author, image, likes, text, id],
    (err, result) => {
        if (err) {
            console.log(err);
            res.status(500).send('Internal Server Error');
        } else {
            console.log(result);
            res.status(200).send('Resource Updated');
        }
    }
);
});

// Route for update likes of a post
app.put("/api/blogs/like/", (req, res) => {
    const id = req.body.id;
    console.log("Blog ", id, ": add like.");
    db.query(
        "UPDATE vesper_blogs SET likes = likes + 1 WHERE id=?",
        [id],
        (err, result) => {
            if (err) {
                console.log(err);
                res.status(500).send(err);
            } else {
                console.log(result);
                res.status(200).send(result);
            }
        }
    );
});

// Route for update dislikes of a post
app.put("/api/blogs/dislike/", (req, res) => {
    const id = req.body.id;
    console.log("Blog ", id, ": remove like.");
    db.query(
        "UPDATE vesper_blogs SET likes = likes - 1 WHERE id=?",
        [id],

```

```

        (err, result) => {
            if (err) {
                console.log(err);
                res.status(500).send(err);
            } else {
                console.log(result);
                res.status(200).send(result);
            }
        }
    );
});

```

## appfrontend.js

```

import { useState, useEffect } from "react";
import './App.css';

function App() {
    const [page, setPage] = useState("Home");
    const [cart, setCart] = useState([]);
    const [cartTotal, setCartTotal] = useState(0);

    const addToCart = (item) => {
        const existingItem = cart.find(
            (cartItem) => cartItem.title === item.title
        );

        if (existingItem) {
            existingItem.quantity += 1;
            setCart([...cart]);
        } else {
            setCart([...cart, { ...item, quantity: 1 }]);
        }

        setCartTotal(cartTotal + item.price);
    };

    const removeFromCart = (item) => {
        const itemIndex = cart.findIndex(
            (cartItem) => cartItem.title === item.title
        );
        // If item was not found.
        if (itemIndex === -1) return;

        item = cart[itemIndex];
        if (item.quantity === 1) {

```

```

        const updatedCart = cart.filter(
            (cartItem) => cartItem.title !== item.title
        );
        setCart(updatedCart);
    } else {
        const updatedItem = {
            ...item,
            quantity: item.quantity - 1,
        };
        const updatedCart = [...cart];
        updatedCart.splice(itemIndex, 1, updatedItem);
        setCart(updatedCart);
    }
};

let navStyle = "nav-item nav-link link-success-emphasis px-3";

return (
    <>
    <div className="container pb-3">
        <header className="border-bottom lh-1 py-3">
            <div className="row flexnowrap justify-content-between align-items-center">
                <div className="col-4 text-left">
                    <a className="blog-header-logo text-body-emphasis text-decoration-none" href="">Vesper Flora</a>
                </div>
                <div className="col-4 pt-1">
                </div>
                <nav className="col-4 d-flex justify-content-end align-items-center">
                    <a className={page === "Home" ? "fw-bold " + navStyle : "" + navStyle} href="#" onClick={() => setPage("Home")}>Home</a>
                    <a className={page === "Plants" ? "fst-italic " + navStyle : "" + navStyle} href="#" onClick={() => setPage("Plants")}>Plants</a>
                    <a className={page === "Produce" ? "fst-italic " + navStyle : "" + navStyle} href="#" onClick={() => setPage("Produce")}>Produce</a>
                    <a className={page === "Blogs" ? "fst-italic " + navStyle : "" + navStyle} href="#" onClick={() => setPage("Blogs")}>Blogs</a>
                    <a className={page === "About us" ? "fst-italic " + navStyle : "" + navStyle} href="#" onClick={() => setPage("About us")}>About us</a>
                    {cart.length > 0 ?
                        <button type="button" onClick={() => setPage("Cart")}>
                            <span>{cart.length}</span>
                        </button>
                    }
                
```

```

                <a className="nav-item nav-link
link-success-emphasis px-3" href="#" style={{ whiteSpace: 'nowrap' }}>Cart
({cart.reduce((total, { quantity }) => total + quantity, 0)})</a>
                </button>
                : <a className="nav-item nav-link
link-success-emphasis px-3" href="#" onClick={() => setPage("Cart")}>Cart</a>
            </nav>
        </div>
    </header>

</div>
<main className="container">
{
    (page === "Home" && <Home cart={cart} setPage={setPage} />)
    || (page === "Plants" && <Plants cart={cart} addToCart={addToCart}
removeFromCart={removeFromCart} />)
    || (page === "Produce" && <Produce cart={cart}
addToCart={addToCart} removeFromCart={removeFromCart} />)
    || (page === "Blogs" && <Blogs />)
    || (page === "About us" && <About />)
    || (page === "Cart" && <Cart cart={cart} cartTotal={cartTotal}
setCartTotal={setCartTotal} setCart={setCart} addToCart={addToCart}
removeFromCart={removeFromCart} />)
}
</main>
</> // end app
); // end return
}

function ProductList({ products, cart, removeFromCart, addToCart }) {
    const listItems = products.map((product, index) => {
        // find the product in the cart if it exists
        const productInCart = cart.find(
            (item) => item.title === product.title
        );
        // get the quantity or default to 0 if the product is not in the cart
        const quantity = productInCart ? productInCart.quantity : 0;

        return (
            <div key={index}>
                <div className="col">
                    <div className="row g-0 border rounded overflow-hidden flex-md-col
mb-4 shadow-sm h-md-400 position-relative">
                        <div className="mx-auto d-none d-md-block h-200 text-center">
                            <img src={products[index].image} alt={product.title}
width="200" height="200" preserveAspectRatio="xMidYMid slice" />
                        </div>

```

```

        <div className="col p-2 d-flex flex-column position-static">
            <h4>{product.title}</h4>
            <p className="mb-auto">{product.description}</p>
            <div className="d-flex justify-content-between
align-items-end">
                <div className="btn-group">
                    <button type="button" className="btn btn-sm
btn-outline-secondary" onClick={() => removeFromCart(product)}>
                        -
                    </button>

                    <span className="btn btn-sm
btn-outline-secondary">
                        {quantity}
                    </span>

                    <button type="button" className="btn btn-sm
btn-outline-secondary" onClick={() => addToCart(product)}>
                        +
                    </button>
                </div>
                <p className="text-body-secondary
my-auto">${product.price}</p>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
);
};

return <>{listItems}</>;
}

function Plants({ cart, addToCart, removeFromCart } ) {
    useEffect(() => { getAllPlants(); }, []);
    const [products, setProducts] = useState([]);

    function getAllPlants() {
        fetch("http://127.0.0.1:4000/api/catalog/getPlants")
            .then((response) => response.json())
            .then((data) => {
                console.log("Show Catalog of Plants :");
                console.log(data);
                setProducts(data);
            })
    }
}

```

```

        })
        .catch(error => { console.log(error) });
    }

    return (
        <>
            <div className="mb-2 pt-2 pb-2 mb-md-3 pt-md-5 pb-md-4 border-bottom">
                <strong className="d-inline-block mb-2
text-success-emphasis">Design</strong>
                    <h1>Plants</h1>
                    <div className="text-body-secondary">Catalog | Dec 3</div>
                </div>
                <div className="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-3">
                    <ProductList products={products} cart={cart} addToCart={addToCart}
removeFromCart={removeFromCart} />
                </div>
            </>
        );
    }

    function Produce({ cart, addToCart, removeFromCart }) {
        useEffect(() => { getAllProduce(); }, []);
        const [products, setProducts] = useState([]);

        function getAllProduce() {
            fetch("http://127.0.0.1:4000/api/catalog/getProduce")
                .then((response) => response.json())
                .then((data) => {
                    console.log("Show Catalog of Plants :");
                    console.log(data);
                    setProducts(data);
                })
                .catch(error => { console.log(error) });
        }

        return (
            <>
                <div className="mb-2 pt-2 pb-2 mb-md-3 pt-md-5 pb-md-4 border-bottom">
                    <strong className="d-inline-block mb-2
text-success-emphasis">Produce</strong>
                        <h1>Fresh produce</h1>
                        <div className="text-body-secondary">Catalog | Dec 3</div>
                    </div>
                    <div className="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-3">
                        <ProductList products={products} cart={cart} addToCart={addToCart}
removeFromCart={removeFromCart} />
                    </div>
                </>
            );
        }
    }
}

```

```

        </>
    );
}

function Blogs() {
    const MAX_DISLIKE_NUM = -3;
    const [blogs, setBlogs] = useState([]);
    useEffect(() => { getAllBlogPosts(); }, []);
    const [writeBlog, setWriteBlog] = useState(false);

    useEffect(() => {
        // This effect runs whenever the 'blogs' state changes
        console.log("Blogs state updated:", blogs);
    }, [blogs]);

    function getAllBlogPosts() {
        fetch("http://127.0.0.1:4000/api/blogs/get")
            .then((response) => response.json())
            .then((data) => {
                console.log("Show Blogs :");
                console.log(data);
                setBlogs(data);
            })
            .catch(error => { console.log(error) });
    }

    const blogDelete = async (blog) => {
        console.log("Goodbye. Blog hit maximum dislikes and was voted off.")

        await fetch("http://localhost:4000/api/blogs/delete", {
            method: "DELETE",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ "id": blog.id }),
        })
            .then((response) => response.json())
            .then((data) => {
                console.log("Delete a blog completed : ", blog.id);
                console.log(data);
            });
        getAllBlogPosts();
    }

    const like = (blog) => {
        fetch("http://127.0.0.1:4000/api/blogs/like/", {
            method: "PUT",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify(blog),
        })
    }
}

```

```

        })
        .then((response) => response.json())
        .then((data) => {
            console.log("like completed :)");
            console.log(data);
        });
        blog.likes += 1
    }
    const dislike = (blog) => {
        fetch("http://127.0.0.1:4000/api/blogs/dislike/", {
            method: "PUT",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify(blog),
        })
        .then((response) => response.json())
        .then((data) => {
            console.log("dislike completed >:)"));
            console.log(data);
        });
        blog.likes -= 1
        // dislikes results in a DELETE of the blog post.
        if (blog.likes <= MAX_DISLIKE_NUM && window.confirm('Oh no! Disliked blog is
below maximum dislikes. Remove it?')) {
            blogDelete(blog);
        }
    }
    function toggleLike(id) {
        let updatedList = blogs.map(blog => {
            if (blog.id === id) {
                if (blog.hasOwnProperty('liked')) {
                    if (!blog.liked) {
                        if (blog.disliked) {
                            // Need to double like to cover dislike to like
                            like(blog)
                        }
                        like(blog)
                    } else {
                        dislike(blog)
                    }
                    blog.liked = !blog.liked
                    blog.disliked = false
                    return blog;
                } else {
                    if (blog.disliked) {
                        // Need to double like to cover dislike to like
                        like(blog)
                    }
                }
            }
        })
    }
}

```

```

        like(blog)
        blog.liked = true
        blog.disliked = false
        return blog;
    }
} else {
    return blog;
}
});

setBlogs(updatedList)
}

function toggleDislike(id) {
    let updatedList = blogs.map(blog => {
        if (blog.id === id) {
            if (blog.hasOwnProperty('disliked')) {
                if (!blog.disliked) {
                    if (blog.liked) {
                        // Need to double dislike to cover like to dislike
                        dislike(blog)
                    }
                    dislike(blog)
                } else {
                    like(blog)
                }
                blog.liked = false
                blog.disliked = !blog.disliked
                return blog;
            } else {
                if (blog.liked) {
                    // Need to double dislike to cover like to dislike
                    dislike(blog)
                }
                dislike(blog)
                blog.liked = false
                blog.disliked = true
                return blog;
            }
        } else {
            return blog;
        }
    });

    setBlogs(updatedList)
}
}

const WriteBlogButton = () => {

```

```

        return (
          <>
            <span className="text-center">
              <button type="button" onClick={() => setWriteBlog(true)}
className="btn btn-sm btn-secondary">
                <a className="nav-item nav-link link-success-emphasis px-3"
style={{ whiteSpace: 'nowrap' }} href="#">Blog me</a>
              </button>
            </span>
            <div className="my-5"></div>
          </>
        );
      }

const BlogForm = () => {
  const [title, setTitle] = useState("");
  const [author, setAuthor] = useState("");
  const [text, setText] = useState("");
  async function handleOnSubmit(e) {
    if (!text || !author || !title) {
      alert("Blog not complete.")
      return;
    }
    const newBlogPost = {
      id: blogs.length + 1,
      title: title,
      author: author,
      description: "",
      category: "",
      text: text,
      likes: 0,
    }
    e.preventDefault();
    console.log(e.target.value);
    await fetch("http://127.0.0.1:4000/api/blogs/create", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(newBlogPost),
    })
      .then((response) => response.json())
      .then((data) => {
        console.log("Post new blog completed");
        console.log(data);
      });
    setWriteBlog(false);
    getAllBlogPosts();
  }
}

```

```

        }
        return (
            <>
                <div className="col">
                    <div className="row g-0 border rounded flex-md-col mb-4 shadow-sm position-relative">
                        <div className="p-2 my-1 d-flex flex-column position-static">
                            <div className="row d-flex justify-content-between pb-2 my-2 mx-2 border-bottom">
                                <h4 className="col col-md-6">
                                    <input type="text" placeholder="Title" name="title" value={title} onChange={(e) => setTitle(e.target.value)} />
                                </h4>
                                <div className="col col-md-6 d-flex justify-content-end">
                                    <button type="submit" className="button rounded mr-4 my-2" onClick={handleOnSubmit}>
                                        submit
                                    </button>
                                    <button type="submit" className="button rounded mx-4 my-2" onClick={() => setWriteBlog(false)}>
                                        cancel
                                    </button>
                                </div>
                            </div>
                            <p className="mx-5">
                                <input type="text" placeholder="Author" name="author" value={author} onChange={(e) => setAuthor(e.target.value)} />
                            </p>
                            <p className="text-body-secondary mx-5 mb-3">
                                <textarea className="h-md-400 w-100" type="text" placeholder="Your blog post text here..." name="text" value={text} onChange={(e) => setText(e.target.value)} />
                            </p>
                        </div>
                    </div>
                </div>
            </>
        );
    }

    function BlogPosts({ blogs }) {
        function BlogPost({ blogs, blog, index, toggleLike, toggleDislike }) {
            const [likes, setLikes] = useState(blog.likes);

            const handleLike = () => {
                toggleLike(blog.id);
            }
        }
    }
}

```

```

        setLikes(blogs[index].likes);
    }
const handleDislike = () => {
    toggleDislike(blog.id);

    setLikes(blogs[index].likes);
}
return (
    <div key={index}>
        <div className="col">
            <div className="row g-0 border rounded overflow-hidden
flex-md-col mb-4 shadow-sm position-relative">
                <div className="p-2 my-1 d-flex flex-column
position-static">
                    <div className="row d-flex justify-content-between
pb-2 my-2 mx-2 border-bottom">
                        <h4 className="col col-md-6">{blog.title}</h4>
                        <div className="col col-md-6 d-flex
justify-content-end">
                            <p className="text-body-secondary
my-auto">{likes}</p>
                            <div className="btn-group mb-1 mx-2">
                                <button type="button"
className={blog.disliked ? "btn btn-sm btn-secondary" : "btn btn-sm
btn-outline-secondary"} onClick={handleDislike}>
                                    
                                </button>
                                <button type="button"
className={blog.liked ? "btn btn-sm btn-secondary" : "btn btn-sm
btn-outline-secondary"} onClick={handleLike}>
                                    
                                </button>
                            </div>
                        </div>
                        <p className="mx-5">{blog.author}</p>
                        <p className="text-body-secondary mx-5
mb-3">{blog.text}</p>
                    </div>
                </div>
            </div>
        </div>
    );
}

return blogs.map((blog, index) => {

```

```

        return (
            <BlogPost key={index} blogs={blogs} blog={blog} index={index}
toggleLike={toggleLike} toggleDislike={toggleDislike} />
        );
    });
}

// BLOG PAGE RETURN
return (
    <>
        <div className="mb-2 pt-2 pb-2 mb-md-3 pt-md-5 pb-md-4 border-bottom">
            <strong className="d-inline-block mb-2
text-success-emphasis">Learn</strong>
            <h1>Prompt of the day: How do you care for orchids?</h1>
            <div className="text-body-secondary">Blogs | Dec 12</div>
        </div>
        <div className="row row-cols-1">
            <BlogPosts blogs={blogs} />
            {
                (writeBlog && <BlogForm />)
                || (!writeBlog && <WriteBlogButton />)
            }
        </div>
    </>
);
}

```

```

function About() {
    return (
        <>
            <div className="mb-2 pt-2 pb-2 mb-md-3 pt-md-5 pb-md-4 border-bottom">
                <strong className="d-inline-block mb-2 text-success-emphasis">About
us</strong>
                <h1>Two computer scientists</h1>
                <div className="text-body-secondary pb-3">3 Dec 2023</div>
                <div className="text-body-primary">SE/ComS319 Construction of User
Interfaces, Fall 2023</div>
                <small className="text-body-tertiary">Dr. Abraham N. Aldaco Gastelum -
aaldaco@iastate.edu</small>
            </div>
            <div className="row row-cols-2 g-3 text-center">
                <div className="col">
                    <div className="row g-0 border rounded overflow-hidden flex-md-col
mb-4 shadow-sm h-md-400 position-relative">
                        <div className="pt-3 d-none d-md-block h-200">

```

```

        <img
src={"http://127.0.0.1:4000/images/profile-isaac.jpg"} alt={"Isaac Lo"} width="200"
height="200" preserveAspectRatio="xMidYMid slice" />
    </div>

        <div className="col p-2 d-flex flex-column position-static">
            <h4>Isaac Lo</h4>
            <p>Junior</p>
            <small className="mb-auto">isaaclo@iastate.edu</small>
        </div>
    </div>
</div>
<div className="col">
    <div className="row g-0 border rounded overflow-hidden flex-md-col mb-4 shadow-sm h-md-400 position-relative">
        <div className="pt-3 d-none d-md-block h-200">
            <img src={"http://127.0.0.1:4000/images/profile-sidd.jpg"} alt={"Siddhartha Gudipudi"} width="200" height="200" preserveAspectRatio="xMidYMid slice" />
        </div>

        <div className="col p-2 d-flex flex-column position-static">
            <h4>Siddhartha Gudipudi</h4>
            <p>Sophomore</p>
            <small className="mb-auto">siddu123@iastate.edu</small>
        </div>
    </div>
</div>
</div>
</>
);
}

function Cart({ cart, cartTotal, setCart, setCartTotal, addToCart, removeFromCart }) {
    const [showComponent, setShowComponent] = useState("Checkout")
    const [thisCart, ] = useState([...cart]);
    const [thisCartTotal, ] = useState(cartTotal);
    const [formData, setFormData] = useState({
        firstName: '',
        lastName: '',
        email: '',
        address: '',
        country: '',
        state: '',
        zip: '',
        cardNumber: '',
        expirationDate: '',

```

```

        cvv: '',
    });
const Checkout = (props) => {
    const [values, setValues] = useState({
        firstName: "",
        lastName: "",
        email: "",
        address: "",
        country: "",
        state: "",
        zip: "",
        cardNumber: "",
        expirationDate: "",
        cvv: "",
    });
    const [errors, setErrors] = useState({});

    const handleSubmit = (e) => {
        e.preventDefault();
        const newErrors = {};
        let isValid = true;

        if (!values.firstName) {
            newErrors.firstName = "Please enter your first name.";
            isValid = false;
        }

        if (!values.lastName) {
            newErrors.lastName = "Please enter your last name.";
            isValid = false;
        }

        if (!validateEmail(values.email)) {
            newErrors.email = "Please enter a valid email address.";
            isValid = false;
        }

        if (!values.address) {
            newErrors.address = "Please enter your address.";
            isValid = false;
        }

        if (!values.country) {
            newErrors.country = "Please choose your country.";
            isValid = false;
        }
    }
}

```

```

    if (!values.state) {
      newErrors.state = "Please choose your state.";
      isValid = false;
    }

    if (!validateZip(values.zip)) {
      newErrors.zip = "Please enter a valid zip code.";
      isValid = false;
    }

    if (!validateCardNumber(values.cardNumber)) {
      newErrors.cardNumber = "Please enter a valid credit card number.";
      isValid = false;
    }

    if (!validateExpirationDate(values.expirationDate)) {
      newErrors.expirationDate = "Please enter a valid expiration date
(MM/YY).";
      isValid = false;
    }

    if (!validateCVV(values.cvv)) {
      newErrors.cvv = "Please enter a valid 3-digit CVV number.";
      isValid = false;
    }

    setErrors(newErrors);
    if (isValid) {
      console.log("submitForm();");
      setFormData(values);
      setCart([]);
      setCartTotal(0);
      setShowComponent("Receipt");
    }
  };
  const validateCardNumber = (number) => {
    const re = /^[0-9]{16}$/;
    return re.test(number);
  };
  const validateEmail = (email) => {
    const re = /\S+@\S+\.\S+/;
    return re.test(email);
  };
  const validateZip = (number) => {
    const re = /^[0-9]{5}$/;
    return re.test(number);
  };

```

```

const validateExpirationDate = (date) => {
    const re = /^[0[1-9]|1[0-2]]\d{2}$/;
    return re.test(date);
};

const validateCVV = (cvv) => {
    const re = /^[0-9]{3}$/;
    return re.test(cvv);
};

const handleChange = (e) => {
    setValues({ ...values, [e.target.name]: e.target.value });
};

const handleCreditCardChange = (e) => {
    var newVal = ''
    for (var i = 0, nums = 0; i < e.target.value.length && nums < 16; i++) {
        if (isNumeric(e.target.value[i])) {
            newVal += e.target.value[i]
            nums++;
        }
    }
    e.target.value = newVal
    setValues({ ...values, [e.target.name]: e.target.value });
};

function isNumeric(n) {
    return !isNaN(parseFloat(n)) && isFinite(n)
}

const cardDisplayHandler = () => {
    const rawText = [...values.cardNumber.split('-').join('')].join('');
    const creditCard = []
    rawText.forEach((t, i) => {
        if (i % 4 === 0 && i !== 0) {
            creditCard.push('-');
        }
        creditCard.push(t)
    })
    return creditCard.join('')
}

return (
    <div>
        <div>
            <div className="container">
                <main>
                    <div className="py-2 text-center"></div>

                    <div className="row g-5">
                        <div className="col-md-5 col-lg-4 order-md-last">

```

```

        <h4 className="d-flex justify-content-between align-items-center mb-3">
            <span>Your cart</span>
            <span className="badge bg-primary rounded-pill">
                {props.products.reduce((acc, cur) => {
                    return acc + cur.quantity;
                }, 0)}
            </span>
        </h4>
        <ul className="list-group mb-3">
            {props.products.map((product, index) => (
                <li key={index} className="list-group-item d-flex justify-content-between lh-sm">
                    <div className="d-inline">
                        <h6>{product.title}</h6>
                        <small>{product.quantity}</small>
                    </div>
                    <span className="text-body-secondary">${product.quantity * product.price}</span>
                </li>
            )))
        <li className="list-group-item d-flex justify-content-between">
            <h6>Tax (7%)</h6>
            <span>${(0.07 * thisCartTotal).toFixed(2)}</span>
        </li>
        <li className="list-group-item d-flex justify-content-between">
            <span>Total (USD)</span>
            <strong>${(thisCartTotal + 0.07 * thisCartTotal).toFixed(2)}</strong>
        </li>
    </ul>
</div>
<div className="col-md-7 col-lg-8">

```

```

<h4 className="mb-3">Payment Information</h4>
<form
    className="needs-validation"
    noValidate
    onSubmit={handleSubmit}
>
    <div className="row g-3">
        <div className="col-sm-6">
            <label htmlFor="firstName"
                className="form-label">
                First name
            </label>
            <input
                type="text"
                className="form-control"
                id="firstName"
                name="firstName"
                onChange={handleChange}
                placeholder=""
                value={values.firstName}
                required
            />
            {errors.firstName && (
                <div className="alert
alert-danger">
                    {errors.firstName}
                </div>
            )})
        </div>

        <div className="col-sm-6">
            <label htmlFor="lastName"
                className="form-label">
                Last name
            </label>
            <input
                type="text"
                className="form-control"
                id="lastName"
                name="lastName"
                onChange={handleChange}
                value={values.lastName}
                required
            />
            {errors.lastName && (
                <div className="alert
alert-danger">

```

```

                {errors.lastName}
            </div>
        )}
</div>

<div className="col-12">
    <label htmlFor="email"
className="form-label">
        Email
    </label>
    <input
        type="text"
        className="form-control"
        id="email"
        name="email"
        onChange={handleChange}
        value={values.email}
        placeholder="you@example.com"
        required
    />
    {errors.email && (
        <div className="alert
alert-danger">
            {errors.email}
        </div>
    )}
</div>

<div className="col-12">
    <label htmlFor="address"
className="form-label">
        Address
    </label>
    <input
        type="text"
        className="form-control"
        id="address"
        name="address"
        onChange={handleChange}
        value={values.address}
        placeholder="1234 Main St"
        required
    />
    {errors.address && (
        <div className="alert
alert-danger">
            {errors.address}
        </div>
    )}
</div>

```

```

                </div>
            )}
        </div>

        <div className="col-md-5">
            <label htmlFor="country"
                className="form-label">
                Country
            </label>
            <select
                className="form-select"
                id="country"
                required
                name="country"
                value={values.country}
                onChange={handleChange}>
                >
                <option
                    defaultValue="">Choose...</option>
                <option>United States</option>
                <option>Mexico</option>
                <option>Canada</option>
                <option>Brazil</option>
                <option>Puerto Rico</option>
                <option>India</option>
                <option>Australia</option>
                <option>Philippines</option>
                <option>Taiwan</option>
            </select>
            {errors.country && (
                <div className="alert
                alert-danger">
                    {errors.country}
                </div>
            )}
        </div>

        <div className="col-md-4">
            <label htmlFor="state"
                className="form-label">
                State
            </label>
            <select className="form-select"
                id="state" required name="state" value={values.state} onChange={handleChange}>
                <option
                    defaultValue="">Choose...</option>
                <option>California</option>

```

```

        <option>Iowa</option>
        <option>Nebraska</option>
        <option>Kansas</option>
        <option>Illinois</option>
        <option>Florida</option>
    </select>
    {errors.state && (
        <div className="alert
alert-danger">
            {errors.state}
        </div>
    )}
</div>

<div className="col-md-3">
    <label htmlFor="zip"
className="form-label">
        Zip
    </label>
    <input
        type="text"
        className="form-control"
        id="zip"
        name="zip"
        value={values.zip}
        onChange={handleChange}
        placeholder=""
        required
    />
    {errors.zip && (
        <div className="alert
alert-danger">{errors.zip}</div>
    )}
</div>

<div className="col-lg-12">
    <label htmlFor="cardNumber"
className="form-label">
        Credit card number
    </label>
    <input
        type="text"
        className="form-control"
        id="cardNumber"
        name="cardNumber"
        placeholder="XXXX-XXXX-XXXX-XXXX"
        value={cardDisplayHandler()}
    </input>
</div>

```

```

        onChange={handleCreditCardChange}
        required
    />
    {errors.cardNumber && (
        <div className="alert
alert-danger">
            {errors.cardNumber}
        </div>
    )}
</div>

<div className="col-lg-12">
    <label htmlFor="expirationDate"
className="form-label">
        Expiration Date (MM/YY)
    </label>
    <input
        type="text"
        className={`form-control
${errors.expirationDate ? 'is-invalid' : ''}`}
        id="expirationDate"
        name="expirationDate"
        placeholder="MM/YY"
        value={values.expirationDate}
        onChange={handleChange}
        required
    />
    {errors.expirationDate && (
        <div className="alert
alert-danger">
            {errors.expirationDate}
        </div>
    )}
</div>

<div className="col-lg-12">
    <label htmlFor="cvv"
className="form-label">
        CVV (3-digit)
    </label>
    <input
        type="text"
        className={`form-control
${errors.cvv ? 'is-invalid' : ''}`}
        id="cvv"
        name="cvv"
        placeholder="CVV (3-digit)"
    />

```

```

                value={values.cvv}
                onChange={handleChange}
                required
            />
            {errors.cvv && (
                <div className="alert
alert-danger">
                    {errors.cvv}
                </div>
            )})
        </div>
    </div>
    <hr className="my-4" />
    <button
        className="w-100 btn btn-primary btn-lg"
        type="submit">
        Place Order
    </button>
</form>
</div>
</div>
<div className="py-2 text-center"></div>
</main>
</div>
</div>
</div>
);
};

} // END CHECKOUT

const Receipt = (props) => {
    return (
        <div className="d-flex flex-column mx-5 p-5 gap-5 justify-content-center">
            <h3 className="">Flora Receipt:</h3>
            <div className="list-group">
                <h4 className="d-flex justify-content-between align-items-center
mb-3">
                    <span className="text-primary">Your cart</span>
                    <span className="badge bg-primary rounded-pill">
                        {props.products.reduce((acc, cur) => {
                            return acc + cur.quantity;
                        }, 0)}
                    </span>
                </h4>
                <ul className="list-group mb-3">
                    {props.products.map((product, index) => (
                        <li key={index} className="list-group-item d-flex
justify-content-between lh-sm">

```

```

        <div>
            <h6 className="my-0">{product.title}</h6>
            <small className="text-body-secondary">
                x{product.quantity}
            </small>
        </div>
        <span className="text-body-secondary">
            ${product.quantity * product.price}
        </span>
    </li>
  )})

```

```

<li className="list-group-item d-flex
justify-content-between">
    <h6 className="my-0">Tax (7%)</h6>
    <span className="text-body-secondary">$ {(0.07 *
thisCartTotal).toFixed(2)}</span>
</li>

```

```

<li className="list-group-item d-flex
justify-content-between">
    <span>Total (USD)</span>
    <strong>$ {(thisCartTotal + 0.07 *
thisCartTotal).toFixed(2)}</strong>
</li>
</ul>
</div>

```

```

<h5 className="">Thank you for your purchase.</h5>

<h6>
    <p>{props.formData.firstName} {props.formData.lastName}</p>
    <p>{props.formData.email}</p>
    <p>{props.formData.address}</p>
    <p>{props.formData.country}, {props.formData.state}
{props.formData.zip}</p>
    </h6>
</div>
);
};
} // END RECEIPT

```

```

return (
  <>
    {showComponent === "Checkout" && (
      <Checkout products={cart} />
    )}
    {showComponent === "Receipt" && (

```

```

        <Receipt products={thisCart} formData={formData} />
    )}
</>
);

}

function Home({ setPage }) {
    return (
        <>
            <div className="px-4 py-5 my-5 text-center">
                <h1 className="display-5 fw-bold text-body-emphasis">Plant
pioneers</h1>
                <div className="col-lg-6 mx-auto">
                    <p className="lead mb-4">Quickly design and customize modern,
elegant spaces with Flora plants, the world's most popular flora-first plant equipper,
featuring succulent variables and mixins, a beautiful blog system, extensive prebuilt
plant arrangements, and powerful horticulture insights.</p>
                    <div className="d-grid gap-2 d-sm-flex justify-content-sm-center">
                        <a className="nav-item nav-link link-success-emphasis px-3"
href="#" onClick={() => setPage("Plants")}>
                            <button type="button" className="btn btn-secondary btn-lg
px-4 gap-3">Flora design</button>
                        </a>
                        <a className="nav-item nav-link link-success-emphasis px-3"
href="#" onClick={() => setPage("About us")}>
                            <button type="button" className="btn btn-outline-tertiary
btn-lg px-4">About us</button>
                        </a>
                    </div>
                </div>
            </div>
        </>
    );
}

export default App;

```

END CODE VERBATIM

# Final Comments

Dear Prof. Aldaco or reviewer,

Thank you for taking your time to review our project. Sidd and I both enjoyed creating this project, and we hope you also found some pleasure in reviewing the design of it. Thank you for all the help and effort given to our education—It was a great semester, and we are privileged to work with you all. We look forward to continuing to learn how to create and maintain professional web applications with new and relevant technology with the knowledge we gained throughout this semester.

- Isaac

Dear Prof Aldaco or reviewer

Just wanted to say thanks for checking out our project. Isaac and I had a blast working on it. This semester was awesome, and we're excited to keep learning about making cool web apps with new tech. Thanks for all your help!

-Sidd