Project Report – Plant Store

Problem Statement:

The goal is to create a website for selling plants by the plant store. I chose to make a website for a plant store because I like plants and have considered opening a plant store in the past. I think they are one of the most wholesome things you can spend money on, so that is something I want to promote with a website. By developing a user-friendly website, customers can easily browse and purchase plants at any time and from any location, eliminating the need for travel or fixed operating hours. The website can also provide valuable information, such as plant descriptions, care instructions, and images, to help customers to make informed decisions about their purchases.

Goals and Requirement Analysis:

Functional Requirements:

Product Catalog: The website should provide a comprehensive and organized catalog of plants, including relevant details such as plant names, descriptions, prices, and availability.

User Registration and Profiles: The website should offer user registration functionality, allowing customers to create accounts and manage their profiles.

Shopping Cart and Checkout: The website should have a shopping cart feature that allows users to add plants to their cart, review their selections, and proceed to a checkout process.

Non-Functional Requirements:

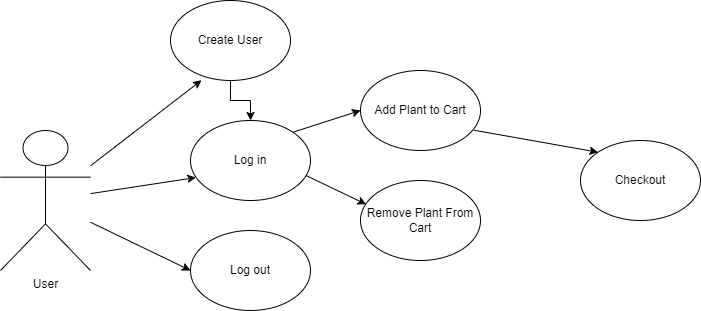
User-Friendly Interface: The website should have an intuitive and user-friendly interface, ensuring that visitors can easily navigate through the different sections, find information, and complete their desired actions.

Responsive Design: The website should be accessible and visually appealing across various devices and screen sizes, including desktops, laptops, tablets, and mobile phones.

Performance and Loading Speed: The website should be optimized for fast loading speeds to ensure a seamless browsing experience.

Security: The website should implement appropriate security measures to protect user information

Use Cases:



Browse and Search Plants:

Stakeholders: Customers, Plant Store

Actors: Website Visitors, Customers

View Plant Details:

Stakeholders: Customers, Plant Store

Actors: Website Visitors, Customers

Add Plants to Shopping Cart:

Stakeholders: Customers, Plant Store

Actors: Customers

Proceed to Checkout:

Stakeholders: Customers, Plant Store

Actors: Customers

Checkout:

Stakeholders: Customers, Plant Store

Actors: Customers

Create User Profile:

Stakeholders: Customers, Plant Store

Actors: Customers

User Interface Specifications

Clear Navigation: The website should have a well-structured navigation menu that allows users to easily browse different sections.

Responsive Design: The website should be designed responsively to ensure a seamless user experience across various devices and screen sizes, including desktops, laptops, tablets, and mobile phones.

Intuitive Layout: The layout should be organized and visually balanced, with a clear hierarchy of information. Important elements, such as search bar, shopping cart, and featured plants, should be prominently displayed.

Visual Imagery: High-quality images of plants should be used to showcase their beauty and appeal.

Product Presentation: Each plant should have a dedicated product page that provides essential information, including the plant's name, description, care instructions, pricing, and availability. Additional details, such as plant dimensions, sunlight requirements, and water needs, can be included as well.

User Registration and Login: The user registration and login process should be straightforward and clearly displayed. Error handling should be implemented to provide informative messages if login or registration fails.

Shopping Cart and Checkout: The shopping cart should display the selected plants, their quantities, and total costs. Users should be able to easily modify the contents of the cart, update quantities, and proceed to a secure checkout process.

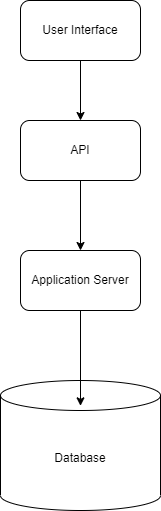
Forms and Input Fields: Forms throughout the website, such as user registration, contact, and payment forms, should be designed to be user-friendly, with clear labels and error messages to assist users in providing accurate information.

Call to Action Buttons: Action buttons, such as "Add to Cart," "Proceed to Checkout," and "Submit," should be visually distinguishable and placed prominently.

Feedback and Confirmation: The website should provide visual feedback and confirmation messages when users perform actions, such as adding items to the cart, submitting forms, or completing a purchase, to ensure a sense of progress and completion.

Accessibility Considerations: The website should adhere to accessibility guidelines, including providing alternative text for images, proper color contrast, etc.

System Architecture:



Client-Side Components:

User Interface (UI): The client-side component comprises the user interface elements, including HTML, CSS (Bootstrap), and JavaScript (jQuery), responsible for rendering and interacting with the website.

Application Layer:

Application Server: The application server handles the business logic of the website. It processes incoming requests, interacts with the database, and generates dynamic content.

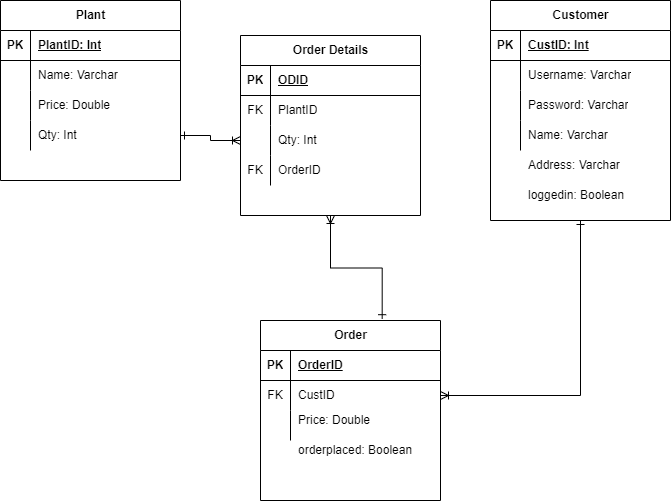
Programming Language/Framework: Java Spring boot

Application Programming Interfaces (APIs): APIs will be developed to provide data and functionality to client-side components.

Database Layer:

Database Management System (DBMS): MySQL

Database Server: The database server handles the storage, retrieval, and management of data.



Plan of Work:

I have 5 weeks from the time of writing until the project due date

Week 1: Technology Evaluation and Selection: There are so many options for developing an e-commerce website. I can only learn so much new technology for this project so I have to decide what to focus on.

Week 2: Design Planning and Database Implementation: This week I want to have a basic design done for the UI with HTML and CSS/Bootstrap, this should help me visualize the project and understand where it is leading. I also want to set up the database in MySQL and get it connecting to my IDE, and make sure that is well designed and functioning properly for my project requirements.

Week 3: Back-end Development: Organize the project into Model, Repository, Service and Controller classes or interfaces. Write the basic CRUD APIs that I will use, and then build the services with more complex business logic. Testing everything with Postman and checking the database to make sure it is functioning as I expect.

Week 4: Front-end Development: Connecting my basic front end design with the APIs that I have written. Fine tuning some elements in the UI. Testing the website with sample users to make sure all of the use cases are working correctly.

Week 5: Testing and Bug Fixing: Checking that everything is working properly. Going through the code to make it more readable and clearer, and simplifying parts where I have repeated myself. Making sure all the required validation is effective.

Problems Faced:

Integrating the front-end with the back-end: Inadequate understanding of the appropriate methods to establish seamless communication between the user interface (front-end) and the server (back-end). This took a lot of work because I really don’t know very much Javascript.

Establishing a connection between the database and the back-end: Insufficient knowledge of the required protocols and procedures to establish a reliable link between the database and the application's server-side.

Navigating Spring Boot: Confronted with the task of acquiring proficiency in utilizing the Spring Boot framework to build and manage the application's back-end infrastructure effectively.

Learning where to implement validation: As my first time building a full-stack application, I wasn’t sure where to put validation for my web services. So, I ended up having some in different places, mostly in the Javascript but some in the Java as well. I’m still not sure what the best-practice is for that, but from what I can tell through testing, the user cannot break the website I built.

Lack of contemporary front-end framework: Unfortunately, I didn’t have time to learn a modern framework such as React or Angular for this project, although I did make use of JQuery. I feel like there is still a lot for me to learn about front-end development, therefore there is lots of room for improvement in that area.

Test Plan:

Step 1: Testing APIs with Postman

In this step, the primary objective is to thoroughly test the APIs using Postman. The testing process involves verifying the functionality and reliability of the web services. Firstly, each API endpoint will be tested to ensure that it can handle various types of requests and respond appropriately. Additionally, the API calls will be checked to confirm that they are correctly interacting with the database. This involves checking if data retrieval, insertion, update, and deletion operations are accurately executed without any unexpected side effects.

A screenshot of a computer

Description automatically generated

Step 2: Testing with Test Users on the Website

Once the API testing is complete, the next step is to evaluate the website's user interface (UI) by testing it with real or simulated users. The focus here is on the end-to-end functionality and usability of the application. Test users, representing typical end-users or specific user personas, will interact with the website as real users would. This testing aims to ensure that the UI is well integrated with the APIs, meaning that user actions trigger the appropriate API calls and handle the responses correctly. Various test scenarios will be executed, covering common use cases and potential user errors, to validate that the website responds accurately and gracefully to user inputs. By conducting these user-centric tests I uncovered UI-related bugs, and made necessary improvements to deliver a user-friendly and satisfying experience.

Future Work:

* Improving the security of the website and the database
* Make the website more flexible for the plant store to add or remove plants as their stock changes
* Add a functioning payment system
* Develop more content for the website (such as plant details, a favourites page, or blog posts from the store about plants)