NAME – SHUBHAM KUMAR **Guide: DR. R. GEETHA RAMANI**

REG. NO: 2019202053 **PROFESSOR(DIST)**

MCA(Regular)

PARADISE NURSERY

ABSTRACT

This project's main aim is to provide to buy plants and their related products, in this developing I'm using react-js for frontend, spring boot in backend for storing data here used MySQL. There are many microservices like React-UI Service, Common Data Service, Authentication Service, Payment Service, Search Suggestion Service. The users can come and buy the plant from our website. There are two types of users. One is seller and the other will be buyer. Seller can come create account as a seller and after creating account they can upload plant image and all the information about the plant for selling. The buyer can view all the plants available for selling on our website. Buyers can read all the details about of the plant and as per their requirement they can select the plant to purchase. If a buyer wants to buy the plant, he/she has to login first and after logging in he/she can buy the plant by providing the delivery address and making payment. Once the booking of the plant is confirm we will start out delivery process to deliver the plant to the buyer address.

INTRODUCTION

Plants are essential to the balance of nature and in people's lives. It is a vital part of the world's biological diversity and an essential resource for the planet. Our goal is to build an online community of buyers and sellers of plants and trees, services and information and to help people provide an outlet from their busy lives & indulge in nature.

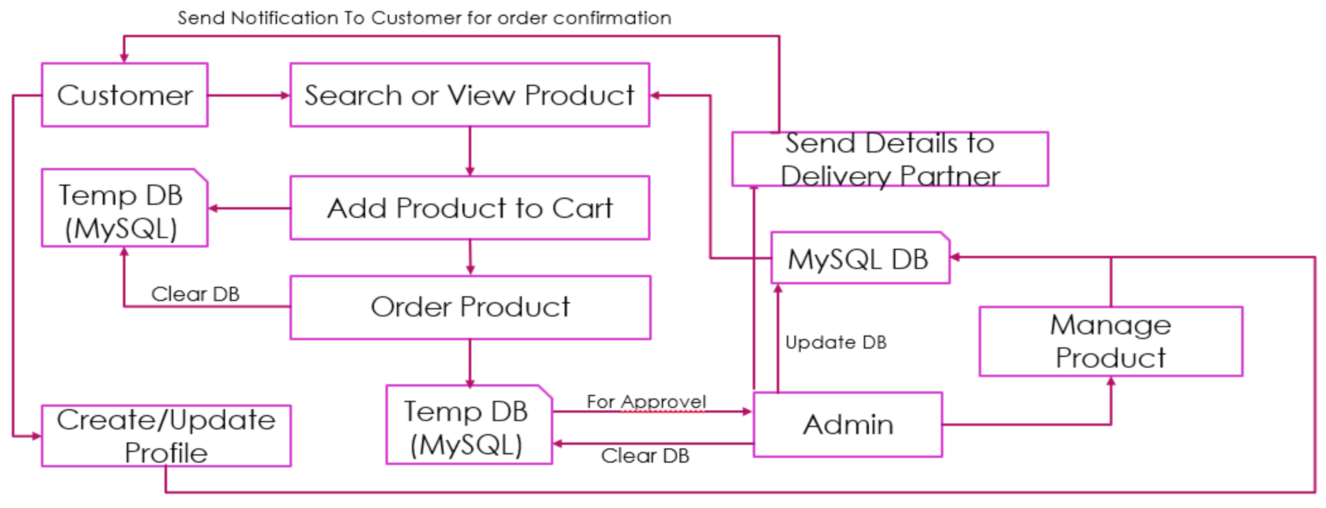
PROBLEM STATEMENT

- Many people want to buy plants and they directly concerned to nursery and plants but sometimes people does not know specific information about particular items as well as seller which are not technically skilled.
- Limited customers reached to the nursery because sometime customer need to travel for long distance as nursery is far from home.

OBJECTIVE:

- Developing a web-based ecommerce for plant that would buy by customer or sell by admin.
- Developing all the services based on microservices
- Deploying our web-based application on AWS

ARCHITECTURE DIAGRAM



EXPLANATION:

Customer visit our web application and search the particular product here using HashMap. And add product to the cart on adding to cart they update the product on temp DB after that on order they send the detail to another DB for verification by admin after verify the detail updated on Main DB and clear all temp DB and send detail to Delivery Partner and also send the confirmation to the customer.

Admin has permission to add or maintain the product in Database.

MODULES:

Admin Module: Add or maintain the product, approve the order after verification

Customer Module: Customer can search, view or buy the product.

Cart: In this customer can add multiple products for order them.

Payment Gateway: It is use for payment the product.

Chat Bot: It provide help to the customer at instant.

FEATURES:

- Google OAuth 2.0 support for quick login. Regular Username/Password authentication.
- Stores user information in the MySQL database.
- Stores API data in Redis Cache to minimize network calls.
- Select filters to display products based on the selections.
- Sort products by popularity, newest, and prices.
- Pagination to display max products on a single page.
- Stores authentication details like token information in cookies.
- Store cart's product information in cookies.
- Payment service using Stripe's API to buy products.
- Responsiveness support for all devices.

TOOLS USED:

- **ReactJS:** Front-end Javascript framework.
- **Spring Boot 2.0:** Back-end JAVA framework to build microservices using Spring Rest Controller and Spring JPA.
- **Material-UI:** Used Google's material design based on the CSS Framework for a responsive website.
- **Semantic-UI:** Used some components which Material-UI doesn't support.
- **MySQL:** Stores product and user information.
- **Redis:** Stores API data in key-value pairs.
- **Cloudinary:** CDN server for storing product images.
- **Google OAuth:** 3rd Party authentication service for quick login by retrieving user profile information.
- **Stripe:** Payment service API to handle user payment requests.
- **AWS Cloud Platform:** Deploying microservices on AWS.
- **Docker-Compose:** Easy way to bring up the application using containerization and behaves similarly in the production environment.

MICROSERVICE:

- **React-UI Service**: Front-end client UI which displays data and makes API calls using Axios API.
- **Common Data Service:** Handles client request to provide common data such as product, filters, categories and order information, etc.
- **Authentication Service:** Creates user account and handles username/password authentication.
- **Payment Service:** Handles payment requests from the client and makes a subsequent request to Stripe API for money deduction.
- **Search Suggestion Service:** Provide default search suggestions and provides suggestions based on a prefix using Hashmap. The service creates the Hashmap based on available data from the database with various combinations and populates the map.