CCL MINI-PROJECT

Title: Urban Garden

Mohib Abbas Sayed 2103158
Hamza Sayyed 2103159
Om Shete 2103163

A

10/11/2022

Description of problem statement

Many people want to buy plants and are directly concerned with the nursery store. But sometimes people do not know specific information about certain plants and the seller does not have technical skills. Build an online nursery store website so that customers can compare prices, view descriptions, and add reviews to a product that helps the customers for a pleasant shopping experience. After the order is placed, an order confirmation report can be viewed by the client for review.

The main objective of this E-Commerce Website project is to create an online nursery store website that helps customers compare prices, view descriptions, can comment on customization ideas, and follow planting tips that promote gardening. Customer service is essential. So, each customer should have a pleasant shopping experience.

1. Offer personalized recommendations for plants based on the customer's preferences and location.
2. Showcase the variety of plants available for purchase.
3. To attract new customers
4. Increase sales by providing an easy-to-use e-commerce platform.
5. Provide helpful resources and guides for gardening enthusiasts.
6. Promote sustainability and eco-friendly practices.

We've containerized our online nursery store project using Docker, encapsulating our application and its dependencies into portable units known as containers. This approach offers flexibility, enabling us to deploy and manage our application consistently across different environments.

For hosting, we've chosen Amazon Web Services (AWS) as our platform and opted to deploy our containers on Amazon Elastic Compute Cloud (EC2) instances. This allows us to have more control over the underlying infrastructure while still benefiting from the scalability and reliability of AWS.

To start, we set up EC2 instances to serve as our hosting environment. We select instance types based on our resource requirements and configure networking and security settings to ensure a secure and accessible environment for our containers.

With our EC2 instances ready, we proceed to deploy our containerized application using Docker. We create and manage Docker containers on these instances, leveraging tools like Docker Compose or Docker Swarm for orchestration.

Deployment is straightforward; we simply SSH into our EC2 instances, pull the Docker images containing our application and run the containers. We can manage multiple containers across different EC2 instances, ensuring high availability and fault tolerance.

To monitor and maintain our application, we utilize AWS CloudWatch for logging and monitoring. CloudWatch allows us to track metrics like CPU utilization and memory usage, set up alarms for automatic notifications, and analyze container logs for debugging purposes.

Overall, by dockerizing our project and hosting it on AWS EC2 instances, we achieve a scalable, reliable, and easily manageable infrastructure. This setup ensures smooth operation of our online nursery store, providing customers with a seamless shopping experience while allowing us to efficiently manage our resources and scale as needed.

Requirement Specification

Understanding the needs of both customers and sellers, we have outlined comprehensive requirements to ensure the successful development of GreenThumb Hub. From user-friendly browsing features to robust backend management tools, our specifications cover all aspects necessary for a seamless online shopping experience.

User Requirements:

- Ability to browse plant varieties: Users should be able to explore a wide range of plant varieties available in the online nursery store.
- Access to detailed plant descriptions including care instructions: Each plant listing should include comprehensive descriptions providing information such as plant species, origin, growth habits, sunlight and water requirements, and care instructions.
- Price comparison feature: Users should be able to compare prices of similar plant varieties offered by different sellers.
- Review system for customers: Users should be able to leave reviews and ratings for purchased plants.
- Order placement and viewing of order confirmation report: Users should be able to easily place orders for selected plants through a seamless checkout process. After placing an order, users should receive a confirmation email containing details of their order.

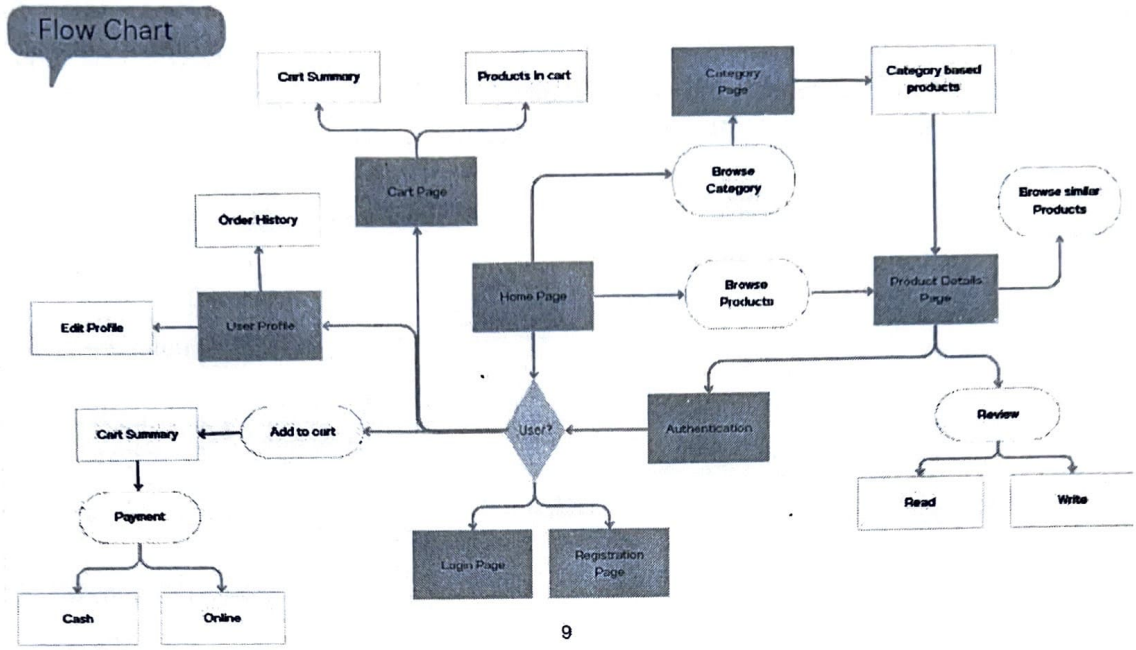
Seller Requirements:

- Easy product management interface: Sellers should have access to an intuitive dashboard for managing their product listings. This interface should allow sellers to add new plant varieties, update descriptions and prices, and remove listings as needed.
- Ability to update plant descriptions and prices: Sellers should be able to edit plant descriptions to provide accurate and up-to-date information to customers. They should also have the flexibility to adjust prices based on market conditions and inventory levels.
- Access to customer reviews and ratings: Sellers should have visibility into customer reviews and ratings for their products. This feedback enables sellers to assess customer satisfaction levels and make improvements if necessary.

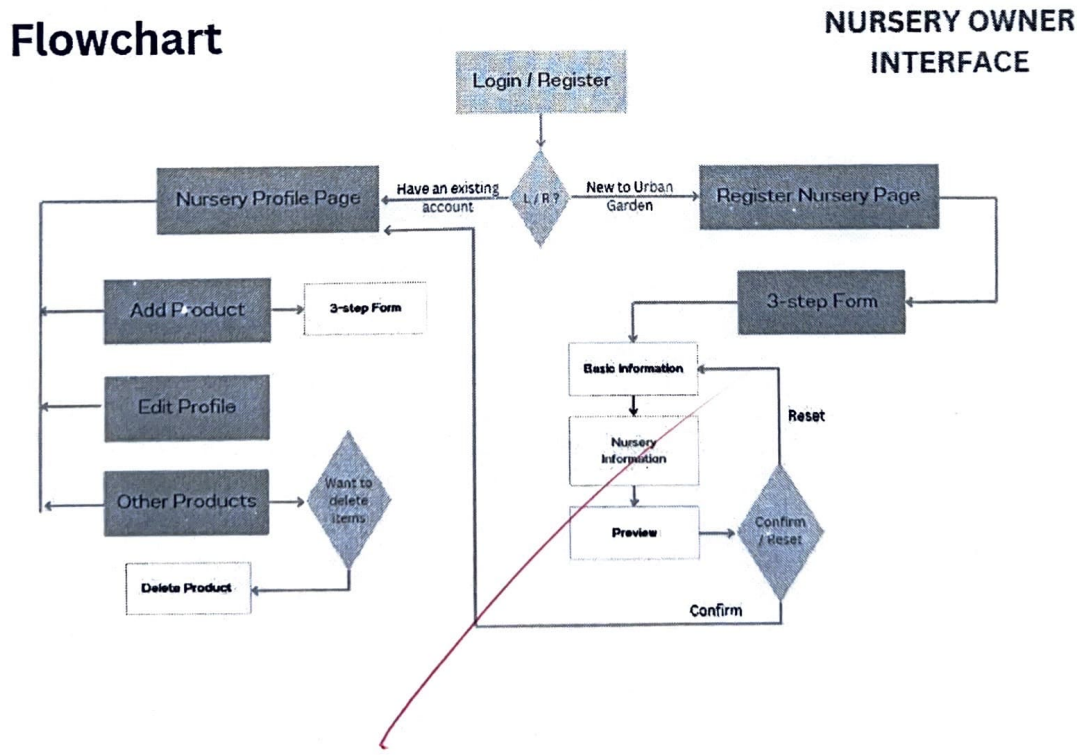
Technical Requirements:

- Dockerized application for easy deployment and scalability: The application should be containerized using Docker to ensure consistency in deployment across different environments. Containerization simplifies deployment and scaling processes, making it easier to manage the application's lifecycle.
- Hosting on AWS EC2 for reliability and scalability: The application should be hosted on Amazon Elastic Compute Cloud (EC2) instances to ensure reliability, scalability, and high availability. EC2 instances can be easily scaled up or down to accommodate changes in demand, ensuring optimal performance for users.
- Integration of payment gateway for online transactions: The application should integrate with a secure payment gateway to facilitate online transactions. This integration enables users to make purchases using credit/debit cards, digital wallets, or other payment methods securely.
- User authentication and account management system: The application should have robust user authentication mechanisms to ensure secure access to user accounts. Users should be able to register for accounts, log in securely, and manage their profiles, including updating personal information and viewing order history.

Block diagram /Architecture Diagram



A visual representation of the architecture illustrates how different components of Urban Graden interact, from the frontend user interface to the backend databases and external services such as payment gateways. This architecture ensures scalability, reliability, and security throughout the system.



Main code/Major steps

Step 1: Install Docker
Install Docker Engine: Go to the official Docker website (https://docs.docker.com/get-docker/) and follow the instructions to download and install Docker Engine for your operating system.

Verify Installation: After installation, open a terminal (or command prompt) and run the following command to verify that Docker is installed correctly:

```
docker –version
```

Step 2: Dockerize Each Component
Create Dockerfile for MongoDB:
Create a file named Dockerfile in your MongoDB directory with the following content:

```
FROM mongo:latest
```

Create Dockerfile for Node.js/Express Backend:
Create a file named Dockerfile in your Node.js/Express project directory with the following content:

```
FROM node:12.19.0

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

ENV PORT=8800
ENV JWT="8hEnPGeoBqGUT6zksxt4G95gW+uMdzwe7EVaRnp0xRI="
ENV MONGO_URL="mongourl"

EXPOSE 8800

CMD [ "npm", "start" ]
```

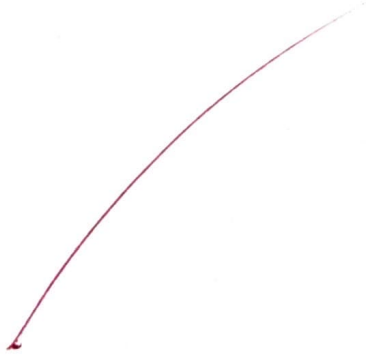
Create Dockerfile for React.js Frontend:
Create a file named Dockerfile in your React.js project directory with the following content:

```
FROM node:14-slim
```

```
WORKDIR /app

COPY ./package.json ./
COPY ./package-lock.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD [ "npm", "start" ]
```

Step 3: Create Docker Compose Configuration
Create a docker-compose.yml file in the root directory of your project:

```
version: "3.8"
services:
  mongodb:
    image: "mongo"
    volumes:
      - data:/data/db
  server:
    build: ./server
    ports:
      - "8800:8800"
    volumes:
      - logs:/app/logs
      - ./server:/app
      - /app/node_modules
    depends_on:
      - mongodb
  client:
    build: ./client
    ports:
      - "3000:3000"
    volumes:
      - ./client/src:/app/src
    stdin_open: true
    tty: true
    depends_on:
      - server
```



volumes:
 data:
 logs:

Step 4: Build and Run Docker Containers
Open a terminal/command prompt in the project directory.

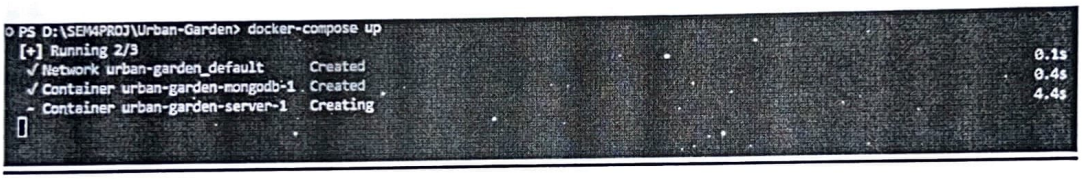
Run the following command to build the Docker images and start the containers:

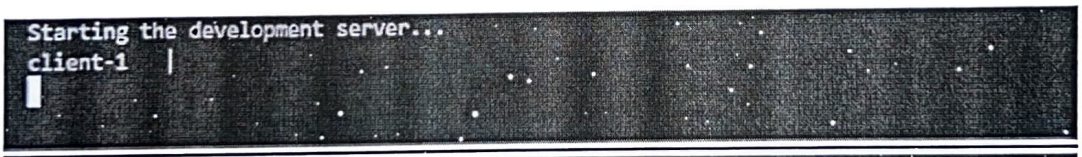
docker-compose up –build

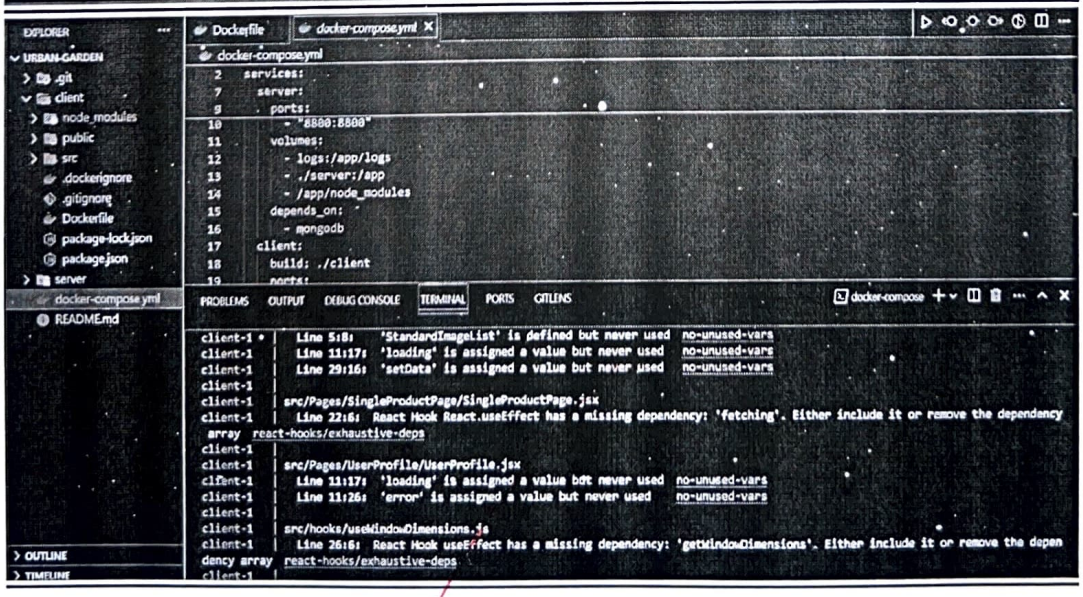
Once the containers are running, you can access your application at http://localhost in your web browser.

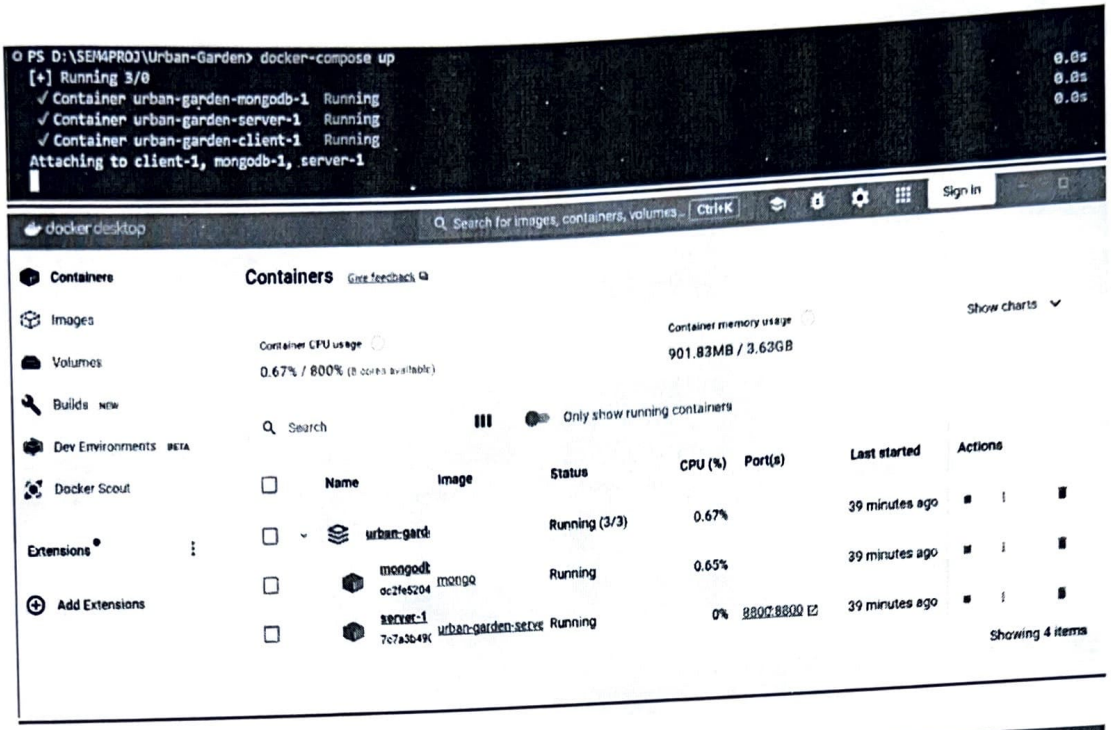
Example of a response from the Server:

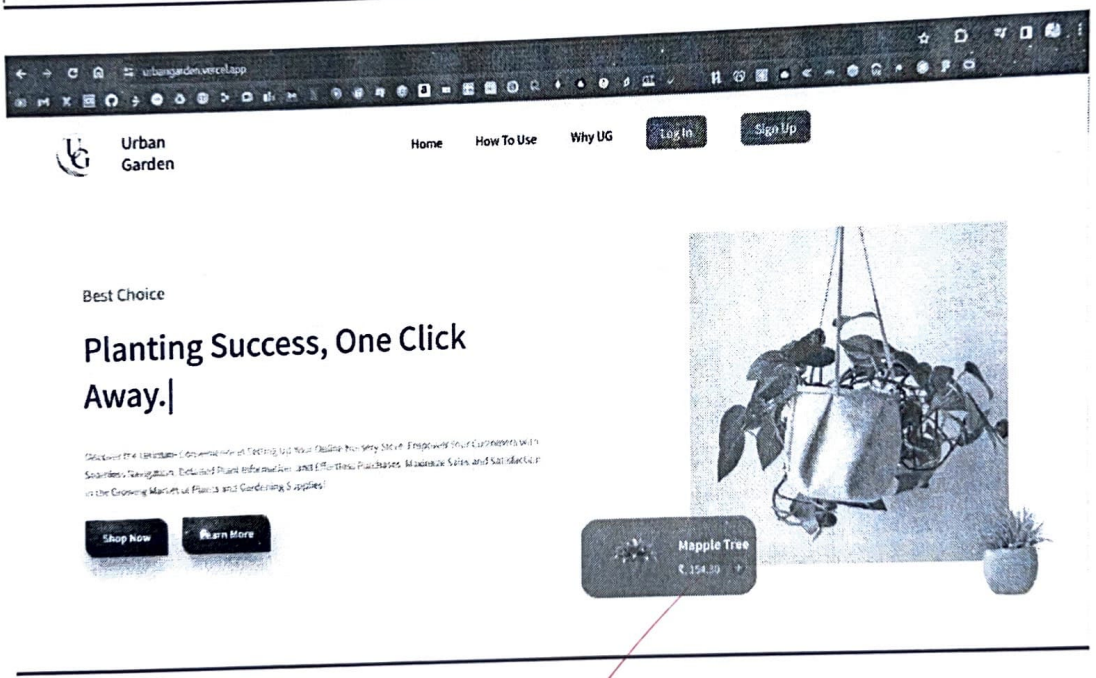
Screenshots of the project in sequence

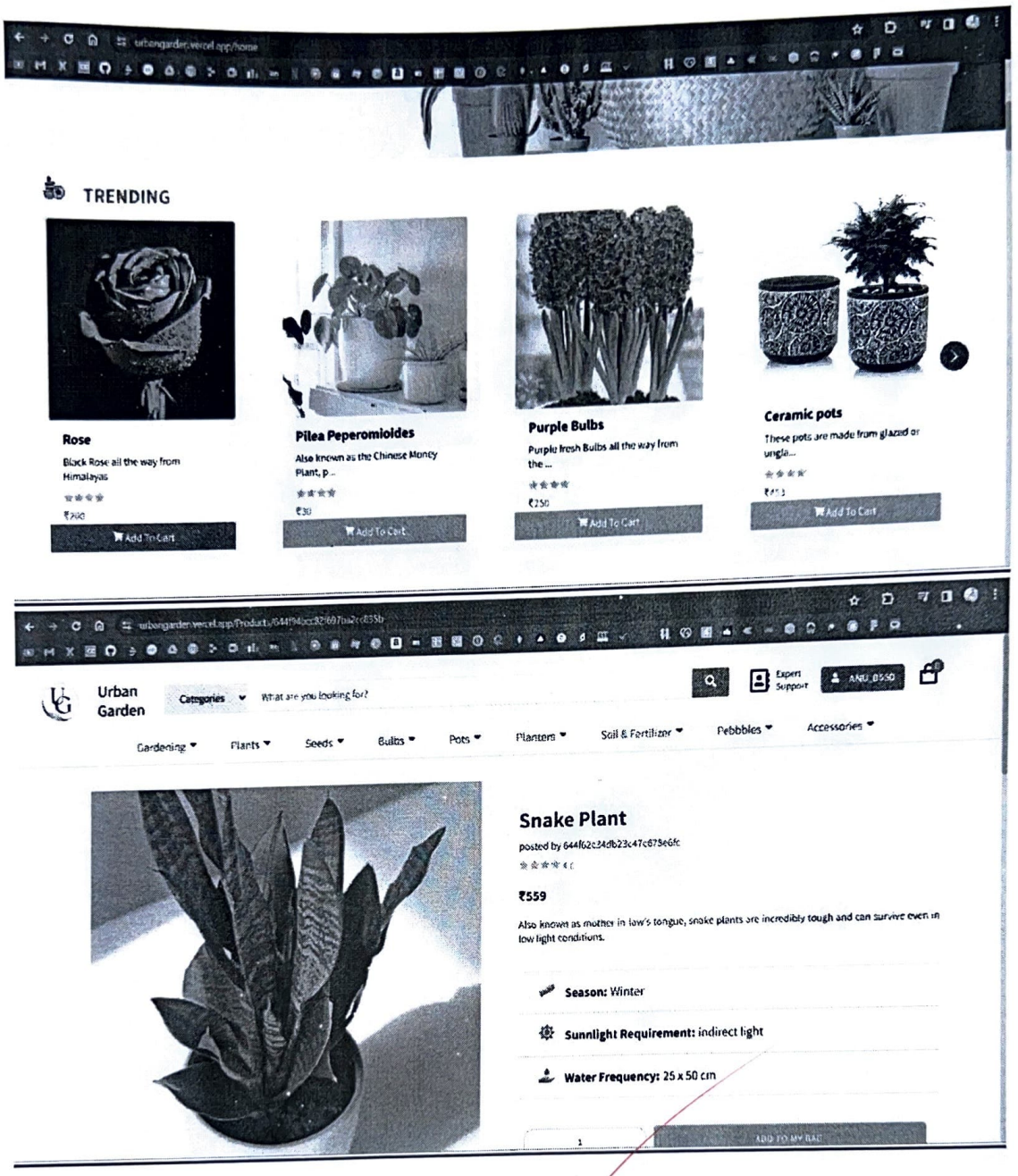


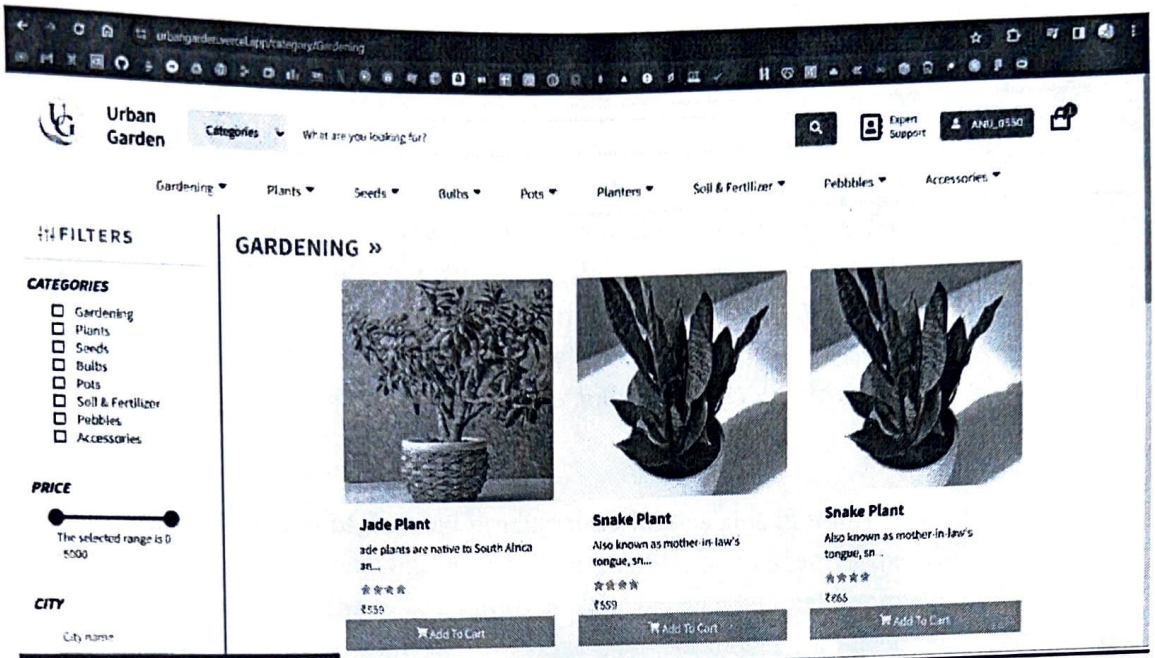


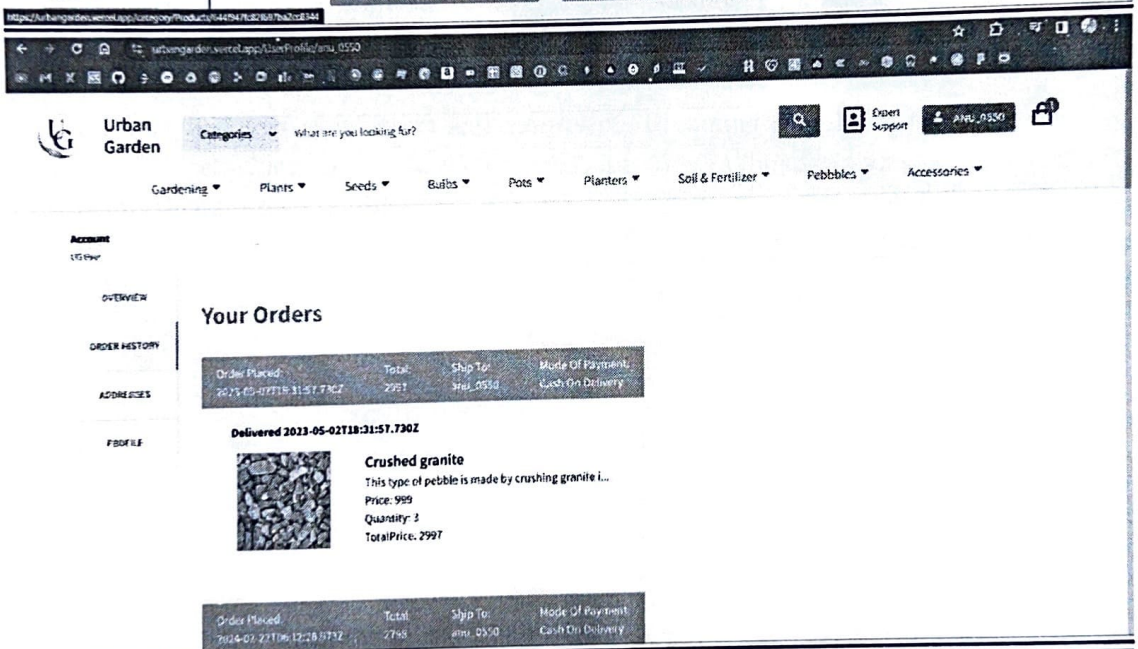


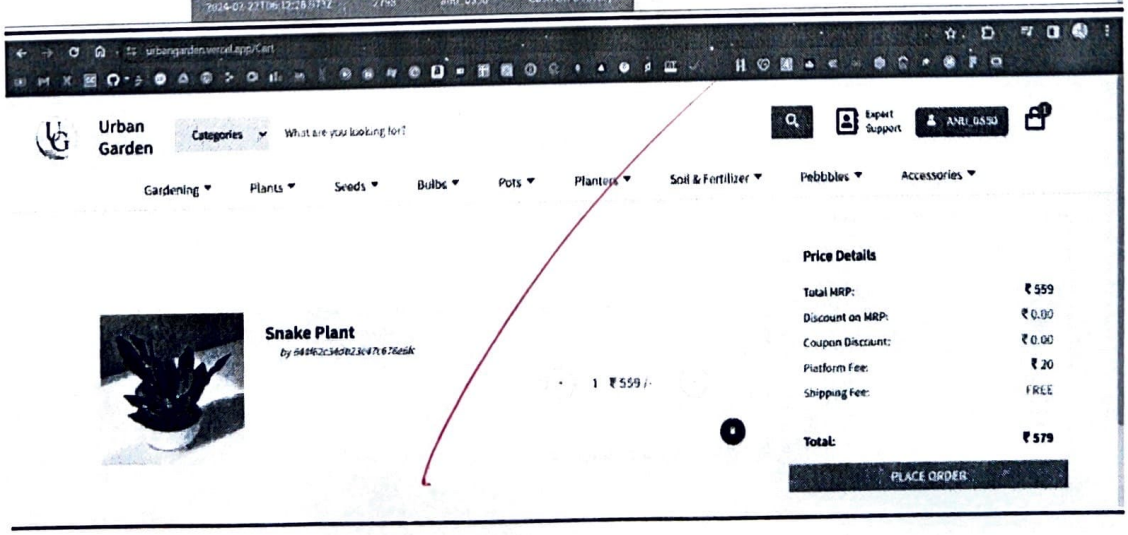

PS D:\SEM4PROJ\Urban-Garden> docker-compose up
[+] Running 3/0
✓ Container urban-garden-mongodb-1 Running
✓ Container urban-garden-server-1 Running
✓ Container urban-garden-client-1 Running
Attaching to client-1, mongodb-1, server-1
0.0s
0.0s
0.0s
docker desktop
Search for images, containers, volumes... Ctrl+K
Sign in
Containers
Images
Volumes
Builds NEW
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Containers Give feedback
Container CPU usage
0.67% / 800% (8 cores available)
Container memory usage
901.83MB / 3.63GB
Show charts
Search
Only show running containers
Name
Image
Status
CPU (%)
Port(s)
Last started
Actions
urban-garde
Running (3/3)
0.67%
39 minutes ago
mongodb
dc2fe5204
mongo
Running
0.65%
39 minutes ago
server-1
7c7a3b49(
urban-garden-serve
Running
0%
8800:8800
39 minutes ago
Showing 4 items


urbangarden.vercel.app
Urban Garden
Home
How To Use
Why UG
Log In
Sign Up
Best Choice
Planting Success, One Click Away.
Shop Now
Learn More
Mapple Tree


urbangarden.vercel.app/home
TRENDING
Rose
Black Rose all the way from Himalayas
Add To Cart
Pilea Peperomioides
Also known as the Chinese Money Plant, p...
Add To Cart
Purple Bulbs
Purple fresh Bulbs all the way from the ...
₹250
Add To Cart
Ceramic pots
These pots are made from glazed or ungla...
Add To Cart
urbangarden.vercel.app/Products/644f94ccc82f697ba2cc855b
Urban Garden
Categories
What are you looking for?
Expert Support
ANU_8550
Gardening
Plants
Seeds
Bulbs
Pots
Planters
Soil & Fertilizer
Pebbbles
Accessories
Snake Plant
posted by 644f62e34db23c47c673e6fc
₹559
Also known as mother in law's tongue, snake plants are incredibly tough and can survive even in low light conditions.
Season: Winter
Sunnlight Requirement: indirect light
Water Frequency: 25 x 50 cm
1
ADD TO MY BAG


Urban Garden
Categories
What are you looking for?
Expert Support
ANU_0550
Gardening
Plants
Seeds
Bulbs
Pots
Planters
Soil & Fertilizer
Pebbbles
Accessories
FILTERS
CATEGORIES
Gardening
Plants
Seeds
Bulbs
Pots
Soil & Fertilizer
Pebbles
Accessories
PRICE
The selected range is 0
5000
CITY
City name
GARDENING »
Jade Plant
ade plants are native to South Africa an...
₹559
Add To Cart
Snake Plant
Also known as mother-in-law's tongue, sn...
₹559
Add To Cart
Snake Plant
Also known as mother-in-law's tongue, sn ..
₹465
Add To Cart


Urban Garden
Categories
What are you looking for?
Expert Support
ANU_0550
Gardening
Plants
Seeds
Bulbs
Pots
Planters
Soil & Fertilizer
Pebbbles
Accessories
Account
OVERVIEW
ORDER HISTORY
ADDRESSES
PROFILE
Your Orders
Order Placed
Total
Ship To
Mode Of Payment
Cash On Delivery
Delivered 2023-05-02T18:31:57.730Z
Crushed granite
This type of pebble is made by crushing granite i...
Price: 999
Quantity: 3
TotalPrice: 2997


Urban Garden
Categories
What are you looking for?
Expert Support
ANU_0550
Gardening
Plants
Seeds
Bulbs
Pots
Planters
Soil & Fertilizer
Pebbbles
Accessories
Snake Plant
1
₹559/-
Price Details
Total MRP: ₹559
Discount on MRP: ₹0.00
Coupon Discount: ₹0.00
Platform Fee: ₹20
Shipping Fee: FREE
Total: ₹579
PLACE ORDER

Conclusion

Blooming Delights emerges as a comprehensive online nursery store, facilitating a seamless shopping experience for gardening enthusiasts. With an extensive range of seeds, plants, tools, and related products, customers are empowered to explore, purchase, and engage with their gardening essentials. The platform not only enables easy comparison of prices but also encourages interaction through product ratings and comments, fostering a sense of community among users.

Moreover, the availability of detailed product descriptions aids in informed decision-making, while the provision of customization ideas and planting tips enhances the overall gardening experience. As we look to the future, our focus remains on continual improvement, with plans to expand product categories, introduce diverse payment modes for added convenience, and implement efficient order-related communication channels such as email or SMS updates.

By prioritizing customer satisfaction and innovation, Blooming Delights is poised to evolve into a premier destination for all gardening needs, catering to the aspirations and passions of green-thumbed enthusiasts worldwide.