1. Project Overview and Objectives

- **Project Name:** Business Website
- **Description:** A responsive business website built using **React.js**.
- **Project Goals:**
    1. Showcase company products, services, and portfolio.
    2. Provide a **mobile-friendly, responsive interface**.
    3. Implement **reusable React components** for maintainability.
    4. Include **interactive features** like cart and contact form.
- **Objectives:**
    1. Learn and implement React.js concepts in a real project.
    2. Ensure accessibility and performance optimization.
    3. Organize code for clarity and scalability.
    4. Document all technical aspects for future reference.

2. Setup and Installation Instructions

1. **Clone the repository:**

```
git clone https://github.com/SGauri24/business-website.git
```

2. **Navigate to the project folder:**

```
cd business-website
```

3. **Install dependencies:**

```
npm install
```

4. **Run the project locally:**

```
npm start
```

- Opens at: `http://localhost:3000`

5. **Build the project for production:**

```
npm run build
```

- Compiles the app into the `build/` folder ready for deployment

3. Code Structure Explanation

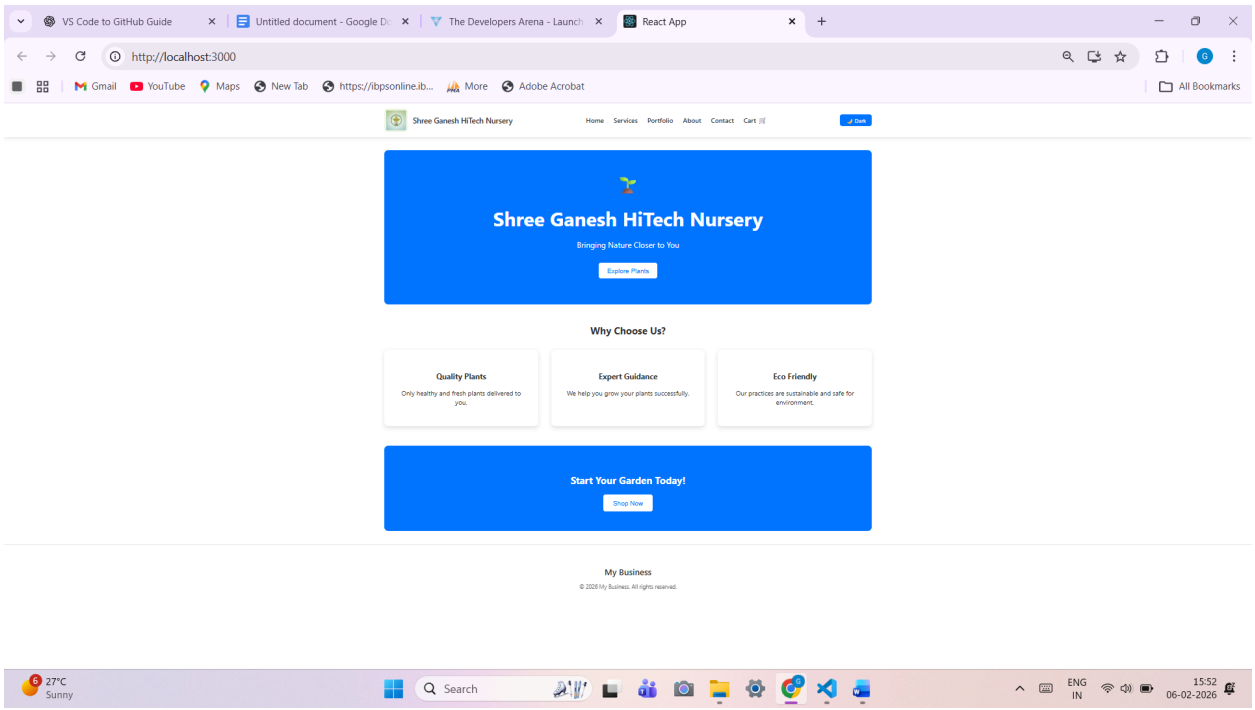
```
business-website/
│
├─ README.md          → Project overview and instructions
├─ .gitignore         → Ignored files (node_modules/, build/)
├─ package.json       → Dependencies and scripts
├─ src/               → React application source code
│   ├─ components/    → Reusable components (Navbar, Footer,
ProductCard)
│   ├─ pages/         → Individual pages (Home, About, Services,
Portfolio, Products, Cart, Contact)
│   ├─ styles/        → CSS (global, responsive, variables,
animations)
│   └─ assets/        → Images, logos, portfolio pictures
├─ public/            → Static assets (index.html, favicon, manifest)
├─ docs/              → Additional documentation (optional)
├─ tests/             → Test files (optional)
```

```
└─ .github/                → GitHub Actions workflows (optional)
```

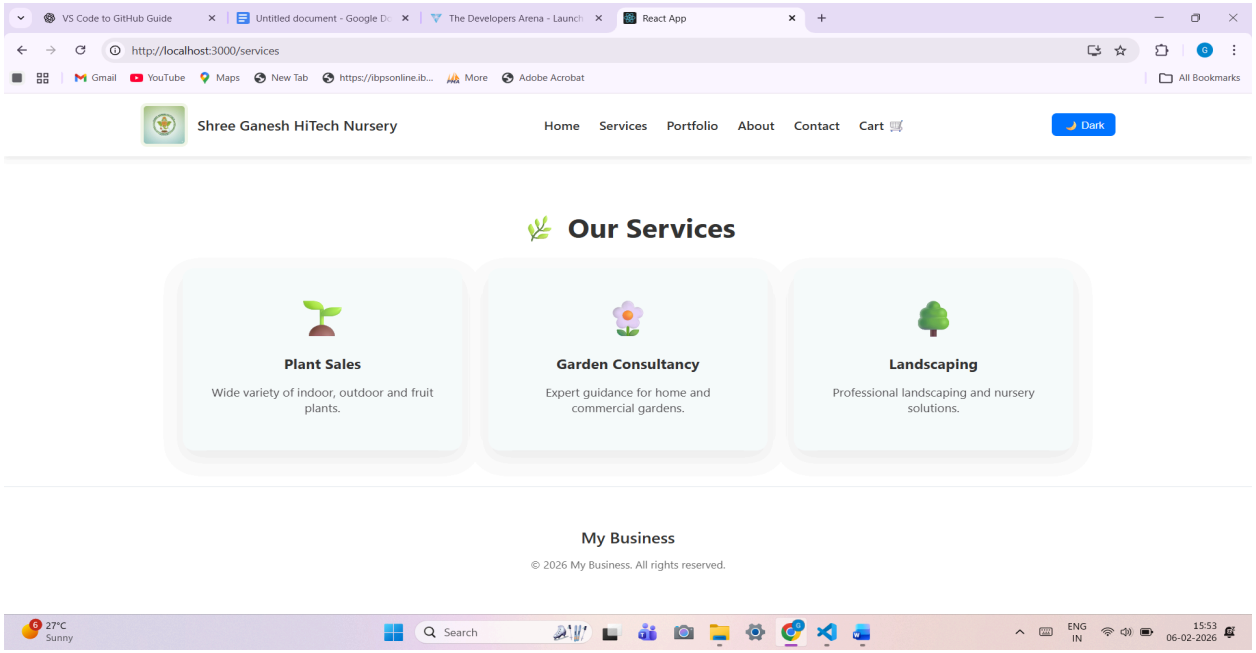
- **Components:** Modular, reusable UI components
- **Pages:** Each route has its own folder
- **Styles:** Organized into global, responsive, and animations for maintainability

4. Visual Documentation (Screenshots)

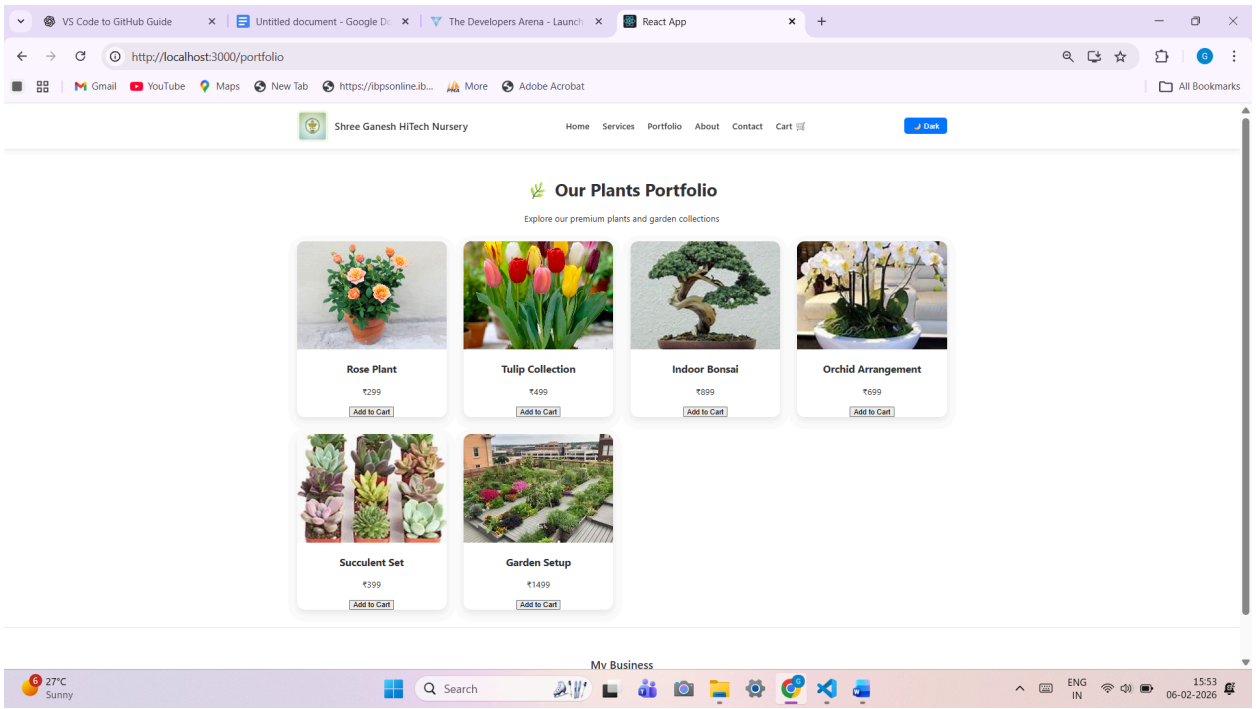
1.Home Page



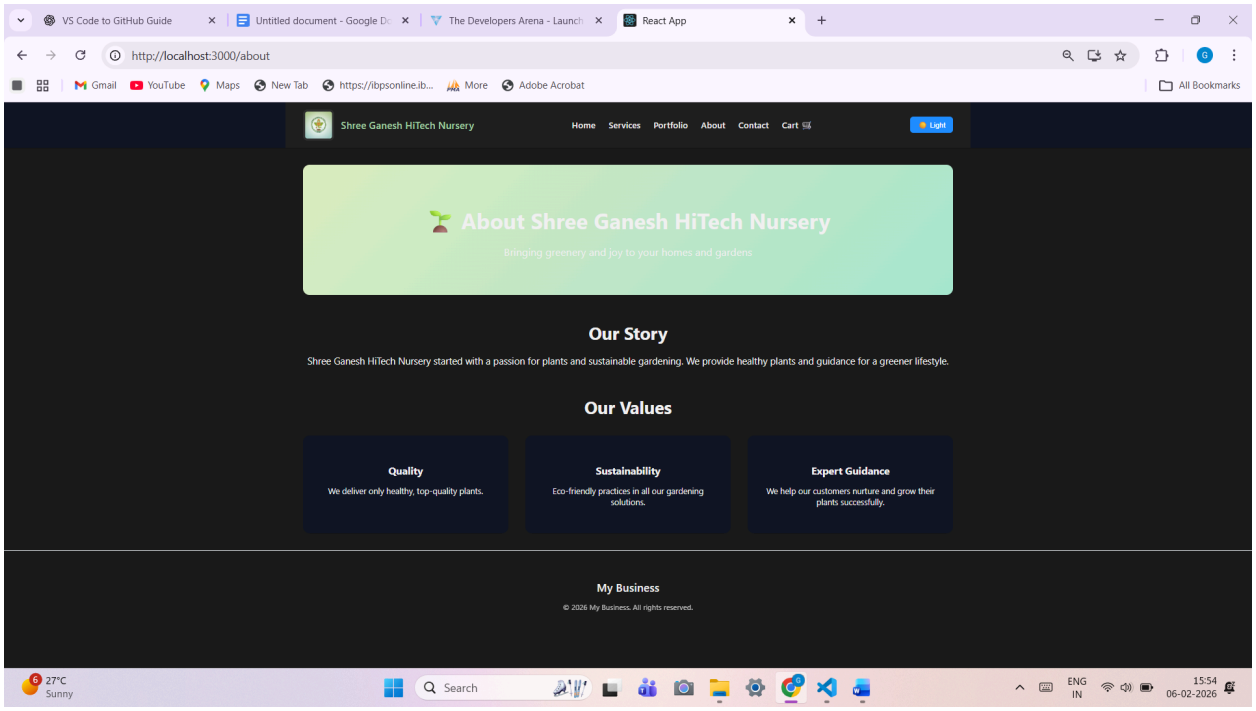
2.Services Page



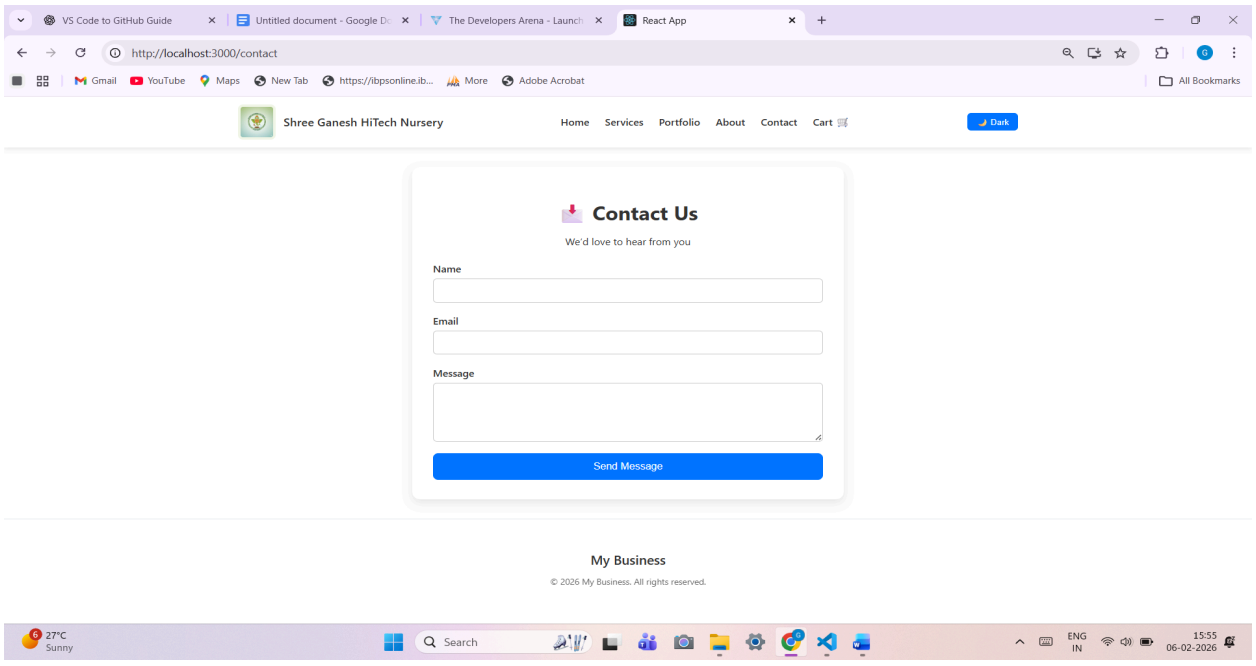
3.Portfolio Page



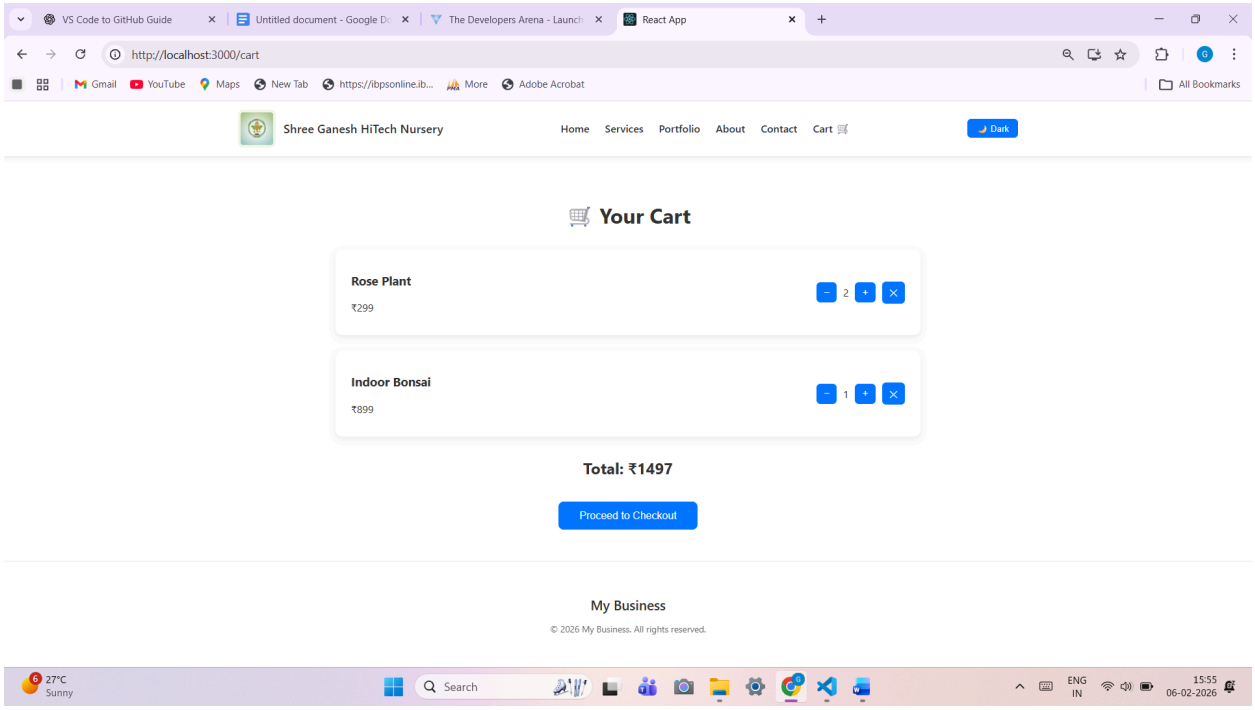
4.About Page



5.Contact Page



6.Cart Page



5. Component Architecture

```
App
├─ Navbar
├─ Footer
├─ Pages
│    ├─ Home
│    ├─ About
│    ├─ Services
│    ├─ Portfolio
│    ├─ Products
│    ├─ Cart
│    └─ Contact
├─ Components
│    └─ ProductCard
```

- Shared components: `Navbar` and `Footer`
- Reusable component: `ProductCard` used in **Products** and **Portfolio** pages.

6. Responsive Strategy

- **Mobile-first design** approach
- **CSS Media Queries** for multiple screen sizes
- **Flexbox** and **CSS Grid** for flexible layouts
- Tested on **desktop, tablet, and mobile**

7. Accessibility Implementation

- Semantic HTML (`<header>`, `<nav>`, `<main>`, `<footer>`)
- **Alt attributes** for all images
- Proper **form labels** for input fields
- **Color contrast** tested for readability
- Focus styles for keyboard navigation

8. Performance Optimization Techniques

- Split code into **reusable components** to reduce re-renders
- Optimized images for faster loading
- Minimized CSS and JS for faster performance
- `.gitignore` prevents unnecessary files from bloating the repo

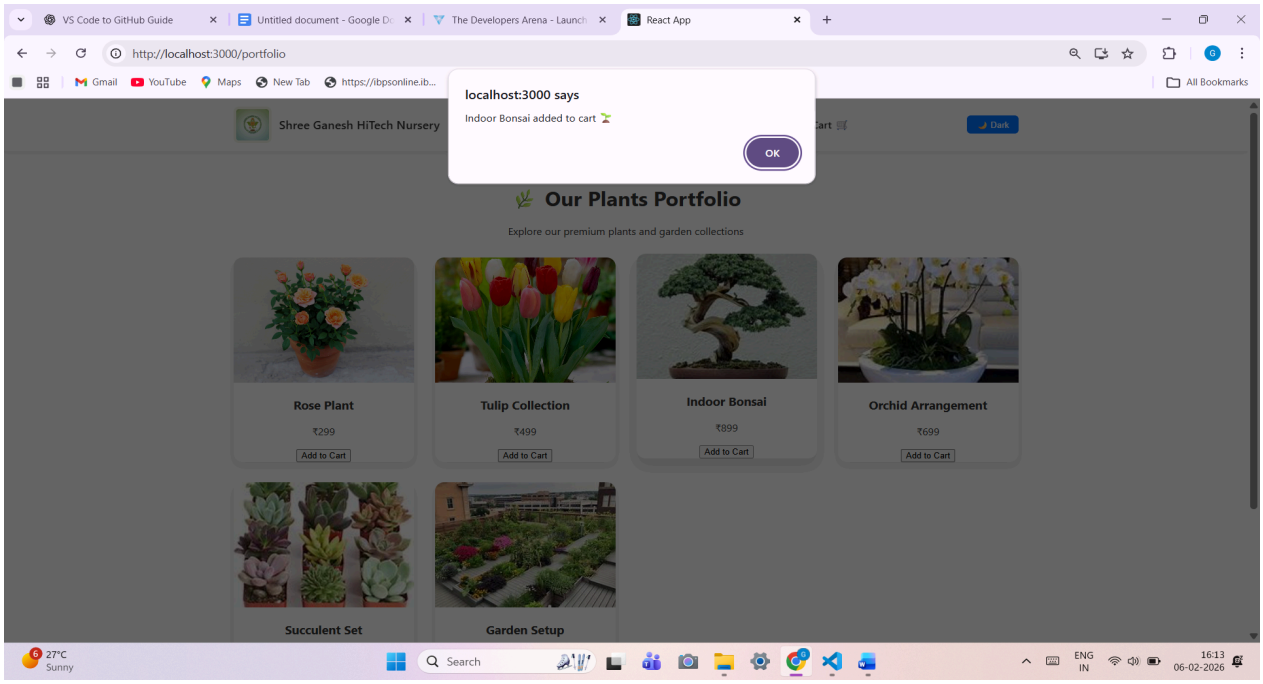
9. Technical Details

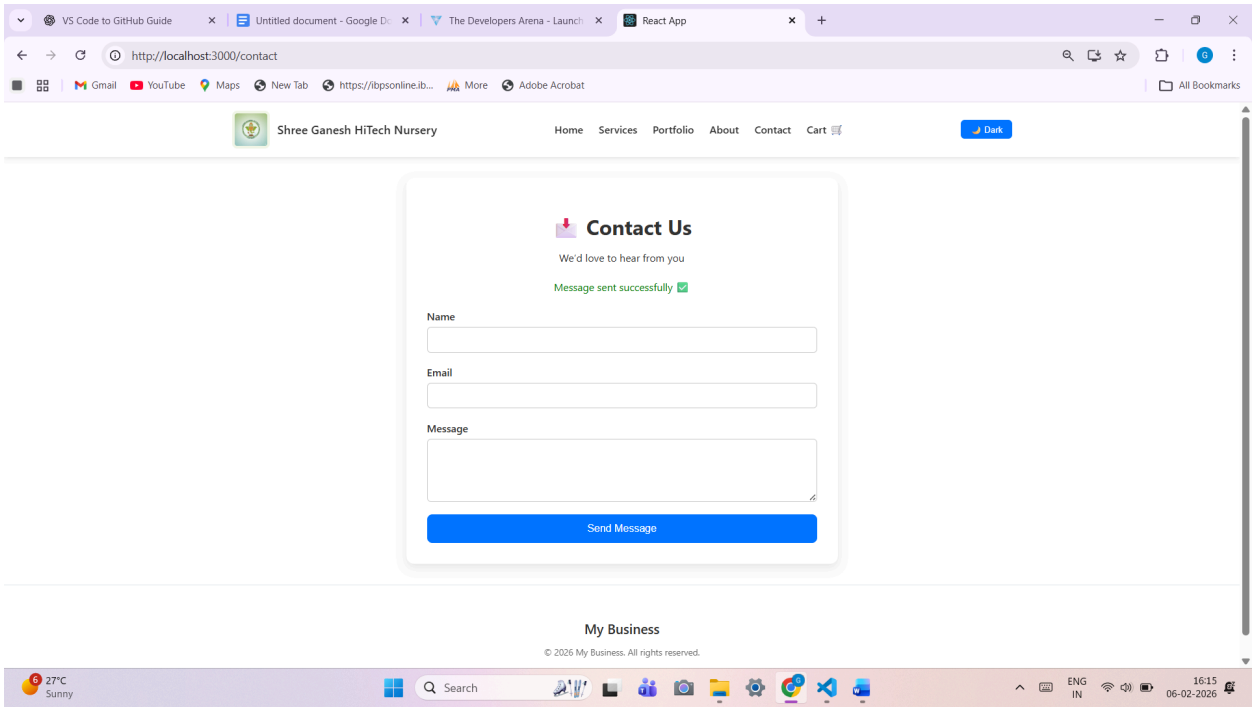
- **Algorithms & Data Flow:**

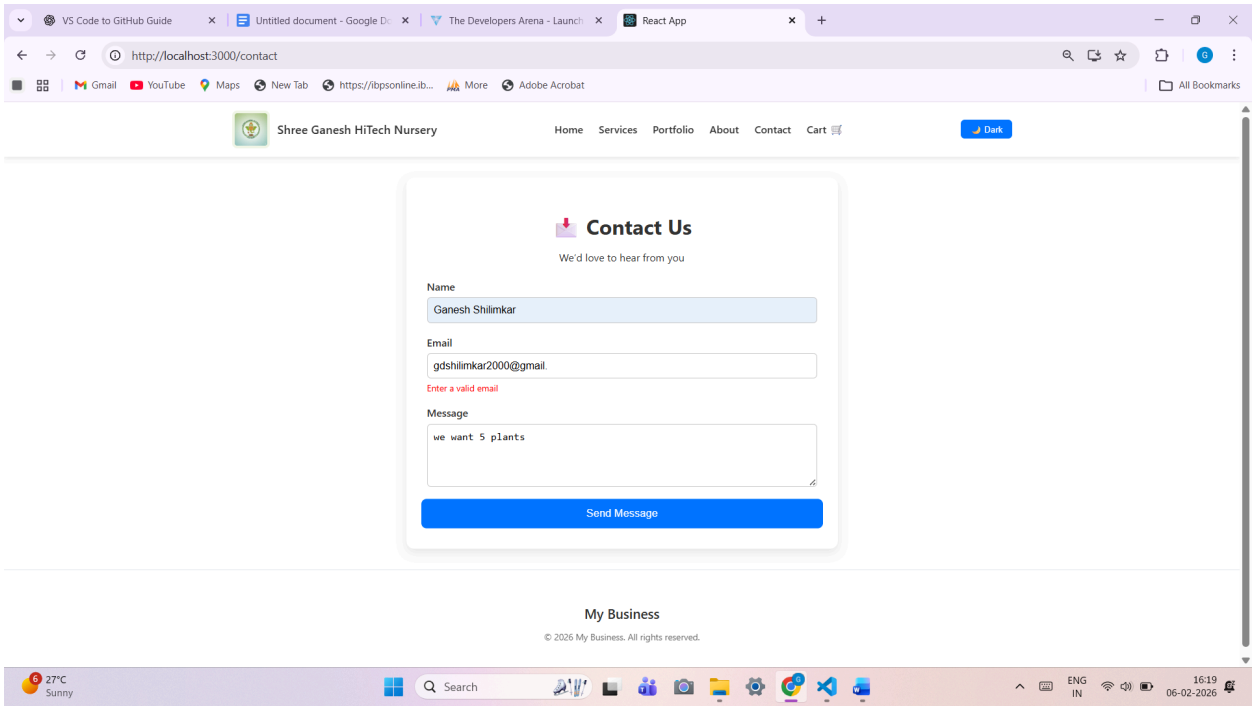
    - Product data flows via **props** into `ProductCard` component
    - State management with **useState** for Cart and other interactive features
- **Architecture:**
    - Functional components for UI
    - CSS modularization for maintainability
    - Component hierarchy ensures clear parent-child data flow

10. Testing Evidence

Feature	Test Case	Expected Result
Navbar Links	Click each link in navbar	Navigate to correct page
ProductCard	Render multiple products	All products/cards displayed correctly
Cart	Add product to cart	Product appears in Cart page
Contact Form	Submit empty form	Validation error messages displayed
Contact Form	Submit valid form	Form submits successfully


VS Code to GitHub Guide
Untitled document - Google Do
The Developers Arena - Launch
React App
http://localhost:3000/portfolio
Gmail
YouTube
Maps
New Tab
https://ibpsonline.ib...
All Bookmarks
localhost:3000 says
Indoor Bonsai added to cart
OK
Shree Ganesh HiTech Nursery
Dark
Our Plants Portfolio
Explore our premium plants and garden collections
Rose Plant
₹299
Add to Cart
Tulip Collection
₹499
Add to Cart
Indoor Bonsai
₹899
Add to Cart
Orchid Arrangement
₹699
Add to Cart
Succulent Set
Garden Setup
27°C
Sunny
Search
ENG
IN
16:13
06-02-2026


VS Code to GitHub Guide
Untitled document - Google Do
The Developers Arena - Launch
React App
http://localhost:3000/contact
Gmail
YouTube
Maps
New Tab
https://ibpsonline.ib...
More
Adobe Acrobat
All Bookmarks
Shree Ganesh HiTech Nursery
Home
Services
Portfolio
About
Contact
Cart
Dark
Contact Us
We'd love to hear from you
Message sent successfully
Name
Email
Message
Send Message
My Business
© 2026 My Business. All rights reserved.
27°C
Sunny
Search
ENG
IN
16:15
06-02-2026



11. Deployment Guide

1. **Build the project:**

   ```
   npm run build
   ```

2. **Host on platforms** like **Netlify**, **Vercel**, or **GitHub Pages**
3. **Upload the `build/` folder** or connect repo directly for automatic deployment
4. Check all **routes, assets, and images** work correctly online

12. How Technical Requirements Were Met

- **React Components:** Modular and reusable
- **Routing:** Each page properly navigable

- **Responsive Design:** Works on all devices
- **Accessibility:** Semantic HTML, alt attributes, labels, color contrast
- **Performance:** Optimized images, minimal re-renders
- **Version Control:** Complete GitHub repository with `.gitignore`