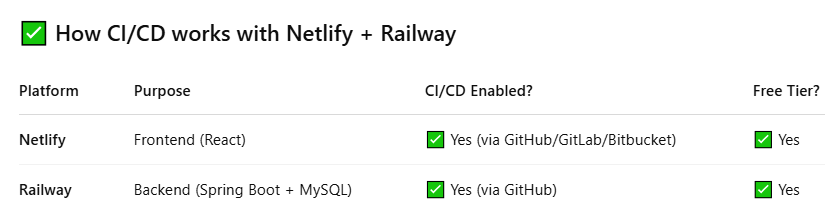
**CI/CD Process USING NETLIFY and RAILWAY**

**CI/CD - (Continuous Integration/Deployment)**

****

### How to Apply CI/CD for Both

#### ✅ For Netlify (Frontend React)

1. **Connect your GitHub repo to Netlify** (one-time setup)
2. Every time you push or merge to a specific branch (e.g. main), Netlify:
   * Builds the updated React app
   * Deploys the new version automatically ✨

🔗 Setup guide:  
**Netlify → Site settings → Build & deploy → Continuous Deployment → GitHub**

#### For Railway (Backend Spring Boot)

1. **Link Railway to your GitHub repo**
2. On every push, Railway:
   * Pulls the updated Spring Boot code
   * Builds and redeploys the backend

🔗 Setup guide:  
**Railway → Project → Settings → Deployments → Connect GitHub repo**

### 🛠️ CI/CD Flow Example

1. You make changes to App.jsx or Controller.java
2. You push to GitHub (git push origin main)
3. **Netlify** rebuilds frontend → deploys to https://yoursite.netlify.app
4. **Railway** redeploys backend → live on your Railway URL

### 🎯 Result

* **No need to manually redeploy**
* Changes go live automatically
* Works even if frontend & backend are hosted on different platforms

**App.jsx (replace with the following code) – (use Backend Railway URL in this file)**

import React, { useEffect, useState } from 'react';

import axios from 'axios';

import './App.css';

function App() {

  const [products, setProducts] = useState([]);

  useEffect(() => {

    axios

      .get(`https://springmysqldeployment-production.up.railway.app/product`)

      .then(response => {

        setProducts(response.data);

      })

      .catch(error => {

        console.error('Error fetching products:', error);

      });

  }, []);

  return (

    <div className="app">

      <h1>📦 Product List</h1>

      <div className="product-list">

        {products.map(product => (

          <div className="product-card" key={product.id}>

            <p><strong>ID:</strong> {product.id}</p>

            <p><strong>Name:</strong> {product.name}</p>

          </div>

        ))}

      </div>

    </div>

  );

}

export default App;

**----------------------------------------------------------------------------------------**

**App.css (replace with the following code)**

.app {

  text-align: center;

  font-family: 'Segoe UI', sans-serif;

  padding: 2rem;

  background: #f0f4f8;

  min-height: 100vh;

}

h1 {

  color: #333;

  margin-bottom: 2rem;

}

.product-list {

  display: flex;

  flex-wrap: wrap;

  justify-content: center;

  gap: 1rem;

}

.product-card {

  background: #ffffff;

  border-radius: 12px;

  padding: 1rem 2rem;

  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

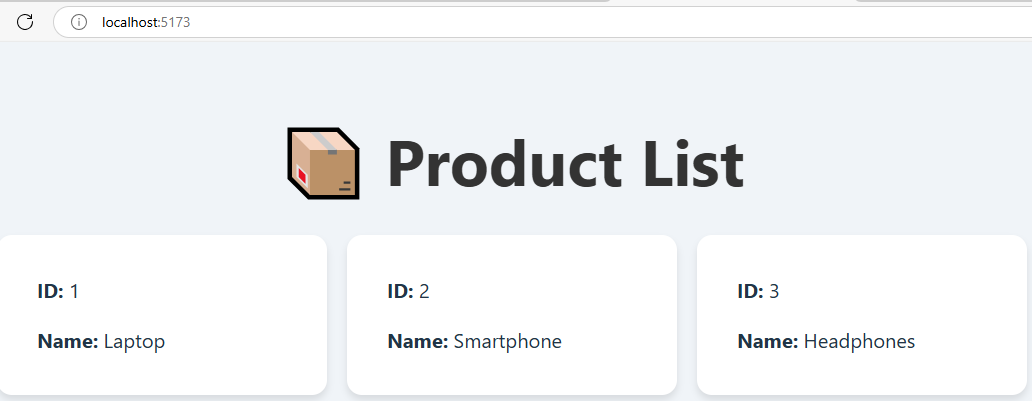
  min-width: 200px;

  text-align: left;

}

**------------------------------------**

**Test locally first: using the command (npm run dev)**

****

**Push this React project to the Github**

**Create a New Repository in the github with the name “FrontendDeployment”**

**Goto command prompt and set the path where is your React project (FrontendDeployment) is available and execute following commands to push project into Github**

****

**Git init**

**Git add .**

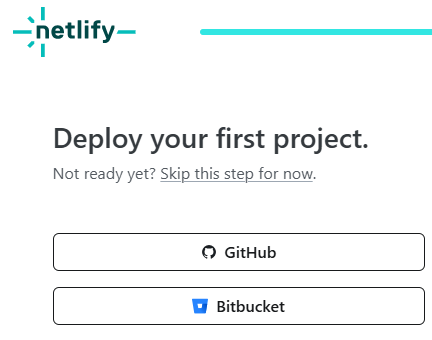
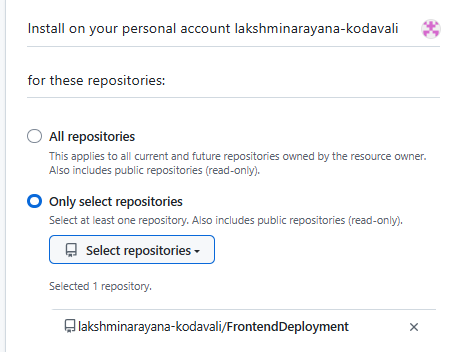
**Git commit –m “ok”**

**Git remote add origin https://github.com/lakshminarayana-kodavali/FrontendDeployment.git**

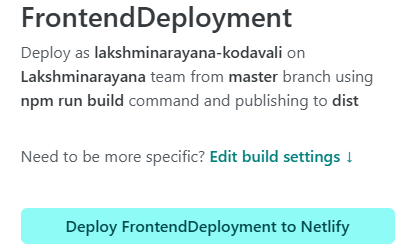
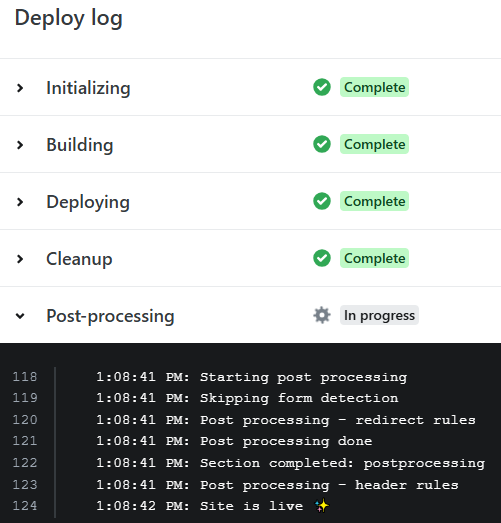
**Git push –u origin master**

**Login into NetlifyApp ( <https://netlify.com/> ) and SignIn**

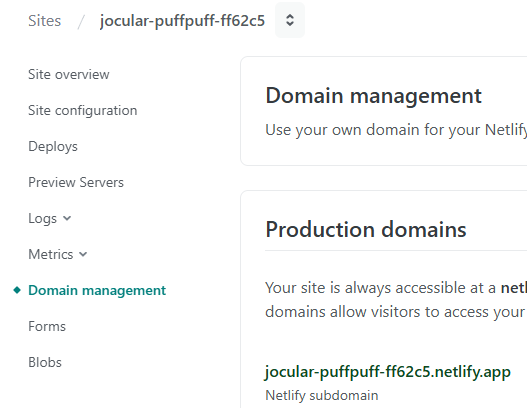
**Click on “Deploy from GitHub repo” (first option) and choose your project to deploy**

** **

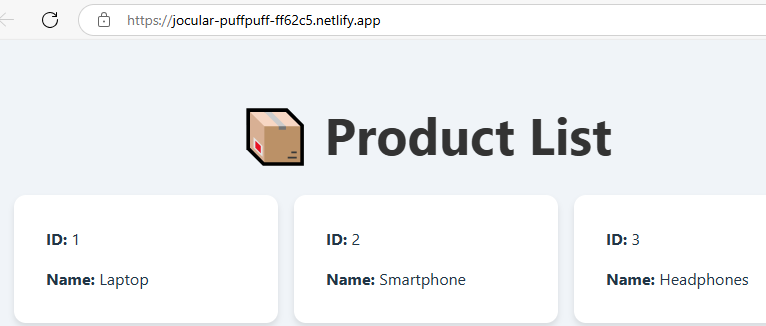
**Click on “Deploy FrontendDeployment to Netlify” and verify the Deployment status (Complete)**

** **

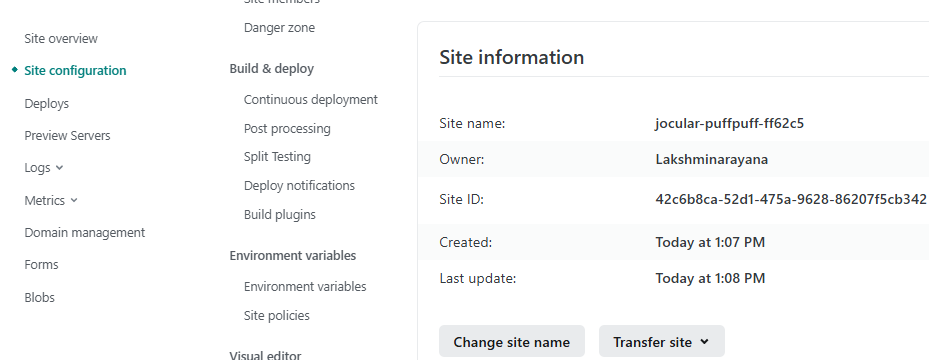
**We can get Netlify URL (Sites -> Domain management )**

****

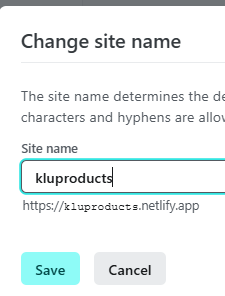
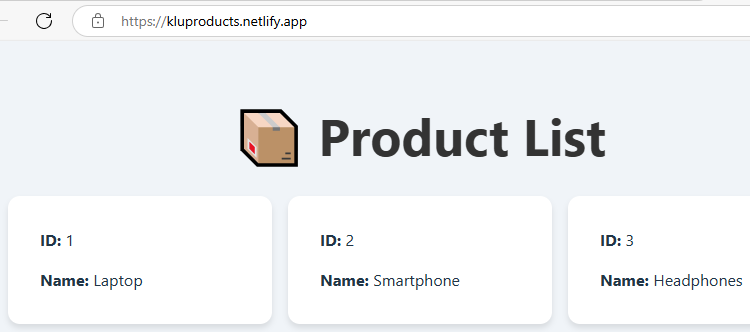
**Click on the Netlify URL to get output**

****

**Customize the URL (Site Configuration -> Change site name)**

****

**Change site name and Run with changed site name (kluproducts) and SAVE it.**

** **