# 🌸 The Petal Pouches

Complete Project Setup Guide

**Document Version:** 1.0

**Date:** October 2025

**Tech Stack:** React.js (JSX) + Node.js + Express + Supabase

## 📋 Project Overview

The Petal Pouches is an Instagram-first e-commerce platform specializing in gifting for teenage girls. The platform features an interactive 3D experience, personalized gift recommendations, and seamless checkout with integrated payments and shipping.

### Frontend

- React.js with JSX
- Vite (Build Tool)
- Tailwind CSS
- 3D Graphics (Three.js)
- Framer Motion

### Backend

- Node.js + Express
- Supabase (PostgreSQL)
- JWT Authentication
- Razorpay/Stripe Payments
- Shiprocket Shipping

## Backend Directory Structure Node.js

```
backend/
│
├── .env                                    # Environment variables (gitignored)
├── .env.example                            # Environment template
├── .gitignore                              # Git ignore rules
├── package.json                            # Dependencies & scripts
├── README.md                               # Backend documentation
│
├── src/
│   ├── index.js                            # Main server entry point
│   │
│   ├── config/                           # Configuration files
│   │   ├── supabaseClient.js               # Supabase client setup
│   │   ├── cloudinary.js                   # Cloudinary config
│   │   ├── razorpay.js                     # Razorpay config
│   │   └── shiprocket.js                   # Shiprocket config
│   │
│   ├── routes/                           # API endpoints
│   │   ├── index.js                        # Main router
│   │   ├── auth.js                         # Authentication routes
│   │   ├── products.js                     # Product CRUD
│   │   ├── orders.js                       # Order management
│   │   ├── cart.js                         # Cart operations
│   │   ├── shipping.js                     # Shipping & tracking
│   │   ├── payments.js                     # Payment webhooks
│   │   ├── bundles.js                      # Bundle routes
│   │   ├── wishlist.js                     # Wishlist CRUD
│   │   ├── reviews.js                      # Reviews & ratings
│   │   └── coupons.js                      # Coupon validation
│   │
│   ├── controllers/                      # Business logic
│   │   ├── authController.js
│   │   ├── productController.js
│   │   ├── orderController.js
│   │   └── ...
│   │
│   ├── services/                         # External API integrations
│   │   ├── supabaseService.js
│   │   ├── cloudinaryService.js
│   │   ├── shiprocketService.js
│   │   ├── razorpayService.js
│   │   └── emailService.js
│   │
│   ├── middlewares/                      # Express middlewares
│   │   ├── auth.js                         # JWT verification
│   │   ├── errorHandler.js                 # Error handling
│   │   └── validator.js                    # Request validation
│   │
│   ├── utils/                            # Helper functions
│   │   ├── helpers.js
│   │   ├── jwt.js
│   │   └── encryption.js
│   │
│   └── db/                               # Database files
│       ├── migrations/                   # SQL migrations
│       └── seeds/                        # Seed data
│
├── scripts/                              # Utility scripts
│   ├── seed_db.js                          # Database seeding
│   └── migrate.js                          # Run migrations
│
├── tests/                                # Test files
│   ├── unit/
│   └── integration/
│
└── logs/                                 # Log files (gitignored)
```

# 📦 Backend Dependencies

## Core Production Dependencies

### express

`^4.18.2`

Fast web framework for Node.js. Handles routing, middleware, and HTTP requests.

### dotenv

`^16.3.1`

Loads environment variables from .env file for secure configuration.

### cors

`^2.8.5`

Enable Cross-Origin Resource Sharing for frontend-backend communication.

### helmet

`^7.1.0`

Secure Express apps by setting various HTTP headers.

### @supabase/supabase-js

`^2.38.4`

Official Supabase client for PostgreSQL database operations.

### jsonwebtoken

`^9.0.2`

Generate and verify JWT tokens for authentication.

### bcryptjs

`^2.4.3`

Hash passwords securely before storing in database.

### cloudinary

`^1.41.0`

Cloud-based image and video management service.

### razorpay

`^2.9.2`

Payment gateway integration for India.

### axios

`^1.6.2`

Promise-based HTTP client for API calls.

## Development Dependencies

### nodemon

`^3.0.1`

Auto-restart server on file changes during development.

### jest

`^29.7.0`

JavaScript testing framework for unit and integration tests.

### eslint

`^8.54.0`

Code linting tool to maintain code quality.

### prettier

`^3.1.0`

Code formatter for consistent styling.

## ⚛ Frontend Directory Structure React.js

```
frontend/
│
├── public/
│   ├── favicon.ico
│   └── assets/
│
├── src/
│   ├── main.jsx                            # App entry point
│   ├── App.jsx                             # Root component
│   │
│   ├── api/                              # Backend API calls
│   │   ├── client.js                       # Axios wrapper
│   │   ├── products.js                     # Product APIs
│   │   ├── orders.js                       # Order APIs
│   │   ├── auth.js                         # Auth APIs
│   │   └── shipping.js                     # Shipping APIs
│   │
│   ├── pages/                            # Route pages
│   │   ├── Home.jsx
│   │   ├── Shop.jsx
│   │   ├── ProductDetail/
│   │   │   └── [id].jsx
│   │   ├── Cart.jsx
│   │   ├── Checkout.jsx
│   │   ├── Account/
│   │   │   ├── Profile.jsx
│   │   │   ├── Orders.jsx
│   │   │   └── Wishlist.jsx
│   │   └── GiftQuiz/
│   │       ├── Quiz.jsx
│   │       └── Results.jsx
│   │
│   ├── components/                       # Reusable components
│   │   ├── layout/
│   │   │   ├── Header.jsx
│   │   │   ├── Footer.jsx
│   │   │   └── Navigation.jsx
│   │   ├── ui/
│   │   │   ├── Button.jsx
│   │   │   ├── Modal.jsx
│   │   │   ├── Input.jsx
│   │   │   └── Card.jsx
│   │   ├── product/
│   │   │   ├── ProductCard.jsx
│   │   │   ├── ProductGrid.jsx
│   │   │   └── ProductFilters.jsx
│   │   ├── cart/
│   │   │   ├── CartItem.jsx
│   │   │   └── CartSummary.jsx
│   │   └── 3d/
│   │       ├── HeroCanvas.jsx
│   │       └── Product3DViewer.jsx
│   │
│   ├── hooks/                            # Custom React hooks
│   │   ├── useCart.js
│   │   ├── useAuth.js
│   │   └── useProduct.js
│   │
│   ├── context/                          # React Context
│   │   ├── AuthContext.jsx
│   │   └── CartContext.jsx
│   │
│   ├── store/                            # Zustand stores
│   │   ├── cartStore.js
│   │   └── authStore.js
│   │
│   ├── lib/                              # Utilities
│   │   ├── api.js
│   │   ├── supabase.js
│   │   └── cloudinary.js
│   │
│   ├── utils/                            # Helper functions
│   │   └── helpers.js
│   │
│   ├── styles/                           # Global styles
│   │   ├── globals.css
│   │   └── tailwind.css
│   │
│   └── config/                           # Configuration
│       └── routes.js
│
├── .env.example
├── .env.local
├── package.json
├── vite.config.js
├── tailwind.config.js
└── README.md
```

# 📦 Frontend Dependencies

## Core Production Dependencies

### react

`^18.2.0`

Core React library for building user interfaces.

### react-dom

`^18.2.0`

React package for working with the DOM.

### react-router-dom

`^6.20.0`

Declarative routing for React applications.

### zustand

`^4.4.7`

Lightweight state management solution.

### axios

`^1.6.2`

Promise-based HTTP client for API calls.

### @supabase/supabase-js

`^2.38.4`

Supabase client for database and auth.

### react-hook-form

`^7.48.2`

Performant form handling with validation.

### yup

`^1.3.3`

Schema validation for form inputs.

### lucide-react

`^0.294.0`

Beautiful and consistent icon library.

### framer-motion

`^10.16.5`

Production-ready animation library for React.

### @react-three/fiber

`^8.15.11`

React renderer for Three.js (3D graphics).

### three

`^0.159.0`

JavaScript 3D library for WebGL graphics.

### tailwindcss

`^3.3.5`

Utility-first CSS framework.

### prop-types

`^15.8.1`

Runtime type checking for React props.

## Development Dependencies

### vite

`^5.0.0`

Fast build tool and dev server.

### @vitejs/plugin-react

`^4.2.0`

Official Vite plugin for React.

### eslint

`^8.54.0`

### prettier

`^3.1.0`

JavaScript linting utility.

Code formatting tool.

# 🚀 Installation Guide

## Backend Setup

```
cd backend npm init -y npm install express dotenv cors helmet @supabase/supabase-js pg jsonwebtoken bcryptjs express-validator cloudinary multer razorpay stripe axios resend twilio uuid dayjs npm install -D nodemon jest supertest eslint prettier
```

## Frontend Setup

```
npm create vite@latest frontend -- --template react cd frontend npm install react-router-dom zustand axios @supabase/supabase-js react-hook-form yup lucide-react framer-motion @react-three/fiber three @react-three/drei lottie-react dayjs uuid prop-types npm install -D tailwindcss postcss autoprefixer npx tailwindcss init -p
```

# 🔐 Environment Variables

### Backend (.env)

```
NODE_ENV=development
PORT=5000
SUPABASE_URL=your_url
SUPABASE_SERVICE_ROLE_KEY=key
CLOUDINARY_CLOUD_NAME=name
JWT_SECRET=secret
RAZORPAY_KEY_ID=key
SHIPROCKET_API_KEY=key
```

### Frontend (.env.local)

```
VITE_SUPABASE_URL=your_url
VITE_SUPABASE_ANON_KEY=key
VITE_API_BASE_URL=http://localhost:5000
VITE_CLOUDINARY_CLOUD_NAME=name
VITE_RAZORPAY_KEY_ID=key
```

# ▶️ Run Commands

## Backend

```
# Development mode with auto-reload npm run dev # Production mode npm start # Run database migrations npm run migrate # Seed database npm run seed
```

## Frontend

```
# Development server (http://localhost:5173) npm run dev # Build for production npm run build # Preview production build npm run preview
```

# ✅ Next Steps

**After Installation:**

1. Configure environment variables in both .env files

2. Set up Supabase database tables (run migrations)
3. Configure Tailwind CSS in frontend
4. Create Supabase and Cloudinary accounts
5. Set up Razorpay/Stripe test accounts
6. Configure Shiprocket API credentials
7. Start both backend and frontend servers
8. Test API endpoints using Postman

📝 **Important Notes:**

- Never commit .env files to version control
- Use different API keys for development and production
- Keep all secrets secure and rotate them regularly
- Test payment integrations in sandbox mode first

Version 1.0 • October 2025