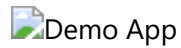









PERN Stack E-Commerce Tutorial



Highlights:

-  Tech stack: PERN + TailwindCSS + DaisyUI
-  Rate Limiting & Bot Detection
-  Global state management with Zustand
-  AI Customer Support Chatbot using Groq LLM
-  Error handling on both server and client
-  Deployment for FREE!
-  And much more!

Getting Started

Setup `.env` file

Create a `.env` file in the `/backend` folder:

```
# Server
PORT=3000

# PostgreSQL
PGUSER=your_pg_user
PGPASSWORD=your_pg_password
PGHOST=localhost
PGDATABASE=your_database_name
PGPORT=5432

# Arcjet (optional)
ARCJET_KEY=your_arcjet_key
ARCJET_ENV=development

# Groq AI LLM API for chatbot
GROQ_API_KEY=your_groq_api_key
```

Installation & Running

Run the Backend API

```
cd backend
npm install
```

```
npm run dev
```





Server will run at: <http://localhost:3000>

Run the Frontend

```
cd frontend
npm install
npm run dev
```

Frontend will run at: <http://localhost:5173>

Notes

-  Ensure **PostgreSQL** is running locally or use a remote DB URL.
 -  The **Groq API key** is required for the AI chatbot. Without it, the chatbot will fallback to a default error message.
 -  Use **npm run seed** (if implemented) to populate initial product data.
 -  For **production deployment**, update **.env** with your production DB and API keys.
-

Tech Stack

Technology	Purpose
PostgreSQL	Database
Express.js	Backend Framework
React	Frontend Framework
Node.js	Runtime Environment
TailwindCSS	Styling
DaisyUI	UI Components
Zustand	State Management
Groq LLM	AI Chatbot
Arcjet	Rate Limiting & Bot Detection

License

This project is open source and available under the [MIT License](#).

Contributing

Contributions, issues, and feature requests are welcome!

Made with  by RishabhMittal

#  Peern Caret- Ecommerce   