

# WhatsApp AI Bot with RAG (Next.js + Vercel Deployment)

---

## 1. Project Overview

Build a production-ready WhatsApp chatbot using:

- Next.js (App Router)
- Twilio WhatsApp Sandbox
- OpenAI API
- RAG (Retrieval Augmented Generation)
- Portfolio-based knowledge system
- Deployment on Vercel

The bot will answer user queries using your portfolio data as context.

## 2. Architecture Flow

User → WhatsApp → Twilio → Next.js API Route (Vercel) → Retrieve Portfolio Chunks (Vector Search) → Send Context + Query to OpenAI → AI Response → Twilio → User

## 3. Tech Stack

- Next.js 14+ (App Router)
- Vercel Deployment
- Twilio WhatsApp Sandbox
- OpenAI API
- Vector Database (Pinecone / Supabase / Chroma)
- Embeddings for RAG

## 4. Step-by-Step Implementation

- Create Next.js project using App Router.
- Create /app/api/webhook/route.ts endpoint (POST).
- Parse Twilio incoming message.
- Create RAG service: load portfolio documents and chunk them.
- Generate embeddings and store in vector database.
- On query: retrieve top similar chunks.
- Send retrieved context + user query to OpenAI.

- Return AI response via Twilio XML response.
- Deploy project to Vercel.
- Set Vercel API URL as Twilio webhook.

## 5. Professional Cursor Prompt (Copy & Paste)

You are a senior full-stack AI engineer.

Build a production-ready WhatsApp chatbot using:

- Next.js 14 (App Router)
- Deployed on Vercel
- Twilio WhatsApp Sandbox
- OpenAI API
- RAG system using portfolio documents

Requirements:

1. Use clean folder structure:  
 /app/api/webhook/route.ts  
 /lib/openai.ts  
 /lib/rag.ts  
 /lib/vector.ts  
 /data/portfolio-docs/
2. Implement RAG:  
 - Chunk portfolio documents  
 - Generate embeddings  
 - Store in vector DB  
 - Retrieve top 3 relevant chunks
3. Send context + query to OpenAI with structured system prompt.
4. AI must:  
 - Only answer from portfolio context  
 - If info not found, respond:  
 "This information is not available in the portfolio."
5. Add proper error handling.
6. Use async/await.
7. Make it production-ready for Vercel.
8. Provide .env.example file.
9. Do NOT create frontend UI.

Make code modular, clean, and scalable.

## 6. Environment Variables (.env Example)

```
OPENAI_API_KEY=your_openai_key
TWILIO_ACCOUNT_SID=your_sid
TWILIO_AUTH_TOKEN=your_token
TWILIO_WHATSAPP_NUMBER=whatsapp:+14155238886
VECTOR_DB_API_KEY=your_vector_db_key
```

---

***End of Advanced Deployment Guide***