

WhatsApp AI Learning Bot - Full Guide

Overview

This guide will help you build a WhatsApp AI Learning Bot that provides free AI resources, research papers

Tech Stack & What You'll Learn

1. FastAPI (Backend Framework)

- Why? Fast, scalable, and async support.
- What You'll Learn? API development, routing, async processing, background tasks, dependency injection

2. Twilio WhatsApp API

- Why? Massive user base, easy integration.
- What You'll Learn? Handling user messages, chat automation.

3. OpenAI/Gemini API

- Why? Generate AI-powered responses.
- What You'll Learn? NLP, text summarization, chatbot responses.

4. MongoDB (Database)

- Why? Flexible NoSQL database with a free tier.
- What You'll Learn? Document-based data storage, indexing, querying.

5. Redis (Caching & Message Queue)

- Why? Speed up responses and manage async tasks.
- What You'll Learn? Caching strategies, handling background tasks.

6. Celery + Redis/RabbitMQ (Background Tasks & Message Queues)

- Why? Handle async tasks (e.g., fetching new AI research daily).
- What You'll Learn? Task queues, distributed processing.

7. Web Scraping (BeautifulSoup/Scrapy)

- Why? Fetch latest AI research papers & news.
- What You'll Learn? Scraping AI-related data sources.

8. Asynchronous Programming (asyncio, httpx, WebSockets)

- Why? Improve efficiency of API requests and real-time interactions.
- What You'll Learn? Async API calls, event-driven architectures.

9. Logging & Monitoring (ELK Stack, Prometheus + Grafana)

- Why? Keep track of errors, usage analytics.
- What You'll Learn? Advanced monitoring & logging.

Implementation Steps

1. Set Up Twilio for WhatsApp
2. Build FastAPI Backend
3. Connect OpenAI/Gemini API for AI Responses
4. Implement MongoDB for Storing User Preferences
5. Implement Web Scraping for AI Resources
6. Use Celery + Redis for Background Tasks
7. Deploy & Monitor with Prometheus + Grafana
8. Expand Features

Next Steps

- Set up WhatsApp API & FastAPI server.
- Integrate AI API & scraping for resources.
- Implement Redis caching and Celery task queues.
- Deploy & optimize performance.
- Expand with advanced backend features.
