TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ Thông tin và Truyền thông

****

**Báo cáo học phần Project III**

Đề tài:

**Xây dựng hệ thống Chatbot tự động**

**với nền tảng Facbook Mesenger hỗ trợ bán hàng**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | Lê Trọng Nhân MSSV: 20173292  Mã lớp: 699564 |
| Giáo viên hướng dẫn: | T.SNguyễn Đình Thuận |
| Bộ môn | Kỹ thuật máy tính |

*Hà Nội, tháng 12 năm 2020*

# MỤC LỤC

[**MỤC LỤC** 1](#_Toc56694504)

[**DANH MỤC HÌNH VẼ** 2](#_Toc56694505)

[CHƯƠNG 1. TÌM HIỂU BÀI TOÁN 3](#_Toc56694506)

[1.1. Messenger Platform 3](#_Toc56694507)

[1.1.1. Cách thức hoạt động 3](#_Toc56694508)

[1.1.2. Webhooks 3](#_Toc56694509)

[1.2. Ngrok 3](#_Toc56694510)

[1.3. Socket.io 4](#_Toc56694511)

[CHƯƠNG 2. NỘI DUNG 4](#_Toc56694512)

[2.1. Các bước thực hiện bài toán 4](#_Toc56694513)

[2.2. Cấu trúc chương trình 5](#_Toc56694514)

[2.2.1. Biểu đồ ca sử dụng 5](#_Toc56694515)

[2.2.2. Biểu đồ trình tự 5](#_Toc56694516)

[2.3. Cài đặt chương trình 6](#_Toc56694517)

[2.3.1. Đăng ký facebook developer 6](#_Toc56694518)

# **DANH MỤC HÌNH VẼ**

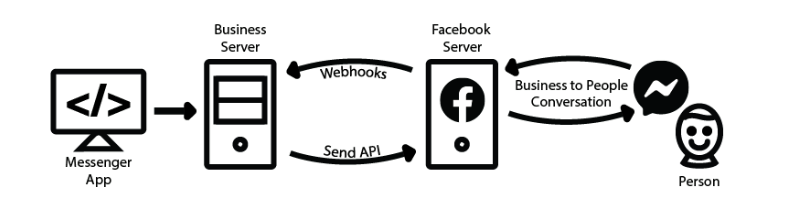
[Hình 1.1 Nền tảng Facebook Messenger 3](#_Toc56628309)

[Hình 2.1 Biểu đồ Use Case chương trình 5](#_Toc56628310)

# TÌM HIỂU BÀI TOÁN

## Messenger Platform

### Cách thức hoạt động



Hình . Nền tảng Facebook Messenger

Trong mô hình nghiệp vụ bán hàng, khi tiến hành cài đặt phần mềm từ bên thứ ba cho những cuộc đối thoại tự động, luồng dự liệu sẽ hoạt động như trên hình 1.1.

Đầu tiên, Facebook sẽ tiến hành gửi thông điệp webhooks cho máy chủ của doanh nghiệp, nơi phần mềm hỗ trợ được cài đặt. Tại đây, phần mềm sẽ xử lý và đưa ra phản hồi tương ứng đến cho Facebook Server qua API được cung cấp. Khi đó, Facebook sẽ phản hồi tin nhắn lại cho người dùng mà ở đây là khách hàng của doanh nghiệp.

### Webhooks

Là một khái niệm ra đời năm 2007, mang đến một cách hữu ích và tương đối dễ dàng, gọn nhẹ trong việc triển khai các phản ứng sự kiện.

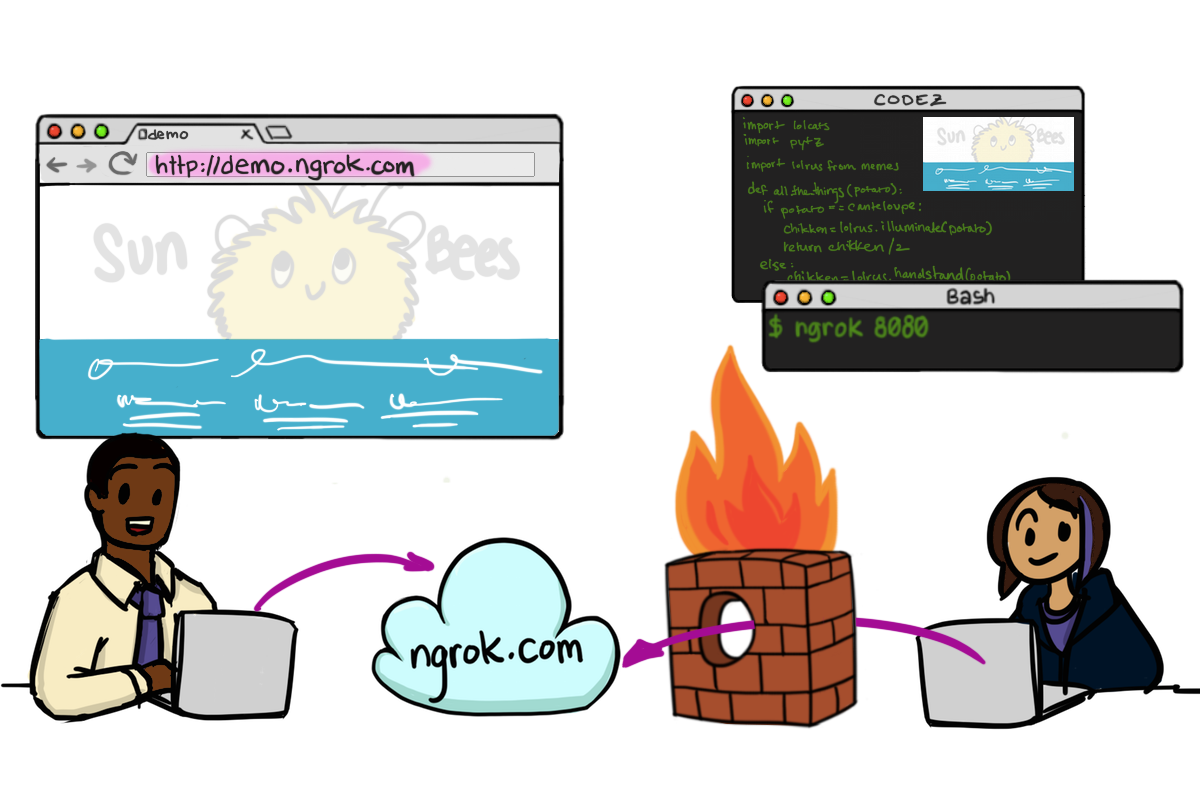
Webhooks đôi khi còn được gọi là "Reverse APIs". Trong các API, ứng dụng client-side sẽ gọi (tiêu thụ) ứng dụng server-side. Trong khi đó, khi có web hook, phía server-side sẽ gọi web hook (end-point URL được cung cấp bởi ứng dụng client-side), ví dụ: ứng dụng server-side gọi ứng dụng client-side [1].

Ưu điểm: Webhooks cho phép người dùng không phải thường xuyên thực hiện các yêu cầu API (API request) để kiểm tra trạng thái của server. Thay vào đó, khi có sự thay đổi hoặc một sự kiện xảy ra, server sẽ gọi đến người dùng qua API để thông báo sự thay đổi đó. => Giảm thiểu được sử dụng băng thông và tải cho server.

Nhược điểm: Khó khăn trong thiết lập ban đầu.

## Ngrok

Là công cụ tạo đường hầm (tunnel) giữa localhost và internet thông qua việc truy cập đến domain đặc biệt của ngrok.



Cách thức hoạt động:

- ngrok tạo kết nối với localhost qua 1 cổng nhất định và tạo 1 url tương ứng với cổng đã được kết nối đó. Khi đó, để ta có thể kết nối với cổng này của localhost qua việc kết nối với trang web public được tạo của ngrok.

## Socket.io

Là một phương thức kết nối cho phép tạo ứng dụng tương tác theo thời gian thực. Các bên sẽ kết nối với nhau qua một server trung gian. Socketio có thể hiểu như một công cụ, hỗ trợ việc lập trình hướng sự kiện (Event Driven Programming), cho phép các bên bắt được các sự kiện với thông điệp đi kèm, từ đó đưa ra hướng xử lý phù hợp.

Trong nội dung của chương trình, Socketio được sử dụng như một server trung gian để kết nối giữa server nhận webhook từ Facebook với ứng dụng trên máy tính.

Socketio có nhiều ưu điểm:

* Hỗ trợ nhiều ngôn ngữ (NodeJS, Java, C++, .NET, Python,…).
* Bảo mật.
* Tự động kết nối cũng như phát hiện ngắt kết nối.
* Hỗ trợ mã hoá nhị phân.
* Hỗ trợ tạo kênh riêng và các phòng riêng để client có thể liên lạc với nhau.
* Dễ cài đặt và sử dụng.

# NỘI DUNG

## Các bước thực hiện bài toán

* Tìm hiểu về các công nghệ sử dụng.
* Thiết kế giao diện cho ứng dụng hỗ trợ chat từ người sở hữu web đến người dùng cần hỗ trợ mà chưa thể xử lý bởi chat bot tự động.
* Thực hiện kết nối giữa ứng dụng với trang web trên fb.
* Cải thiện giao diện.
* Tích hợp NLP cho chatbot.
* Cải thiện hiệu năng và khắc phục lỗi trong ứng dụng.

## Cấu trúc chương trình

### Biểu đồ ca sử dụng

Diagram

Description automatically generated

Hình . Biểu đồ Use Case chương trình

* Người dùng (khách hàng) tiến hành gửi tin nhắn lên hệ thống chat Messenger của facebook.
* Khi hệ thống nhận được tin nhắn, tin nhắn sẽ được lưu vào database.
* Mặt khác, lúc này, tin nhắn sẽ được kiểm tra bởi hệ thống, nếu có thể trả lời được, hệ thống sẽ tiến hành phản hồi tin nhắn, đồng thời lưu tin đã phản hồi vào cơ sở dữ liệu. Trong trường hợp không trả lời được, hệ thống sẽ chuyển tiếp tin nhắn sang ứng dụng mà chủ trang web đang truy cập.
* Chủ trang web phản hồi lại tin nhắn cho người dùng qua hệ thống, đồng thời hệ thống sẽ lưu lại tin nhắn phản hồi này để có thể học tập và trả lời tốt hơn sau đó.

### Biểu đồ trình tự

A picture containing diagram

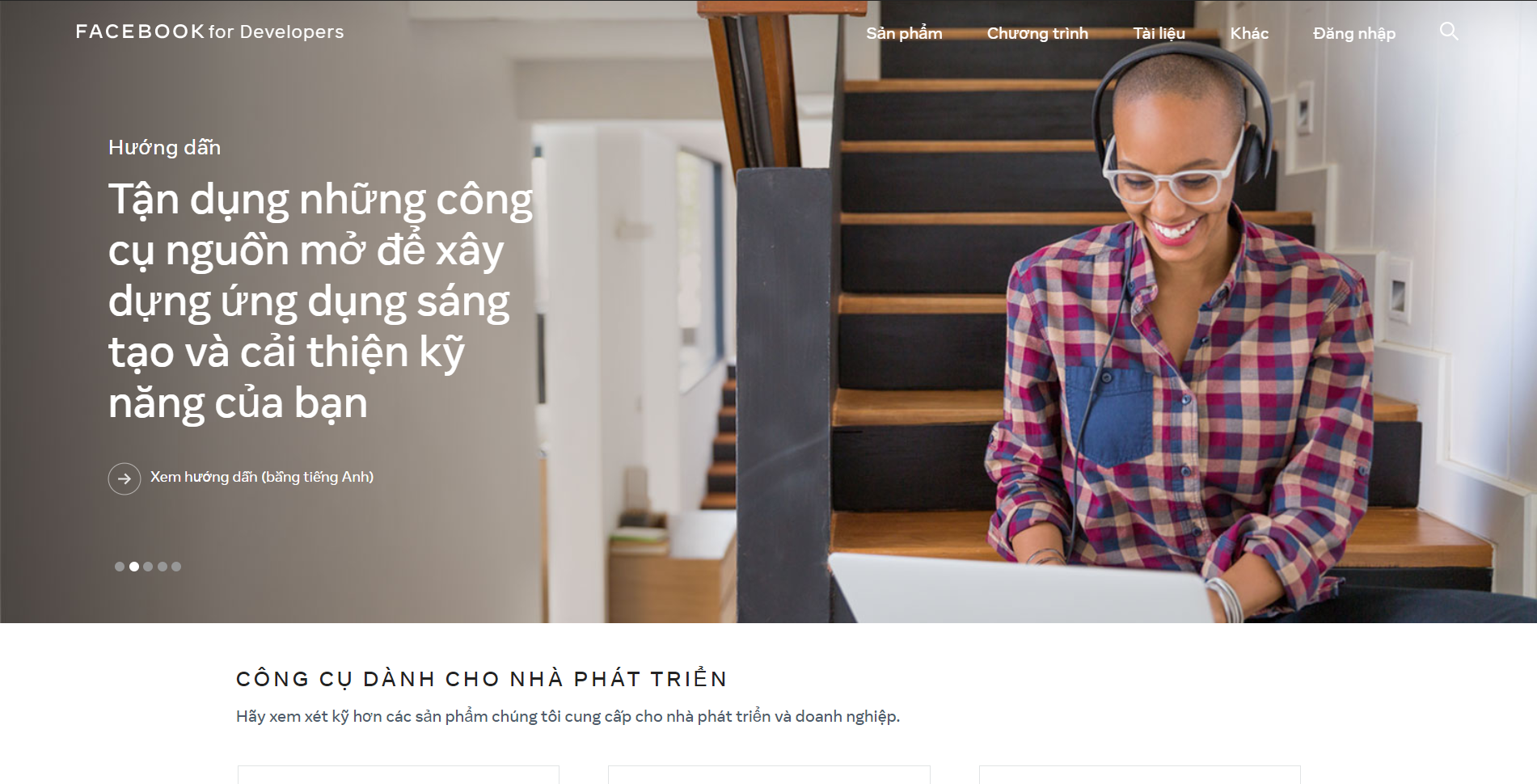
Description automatically generated

Hình . Biểu đồ trình tự

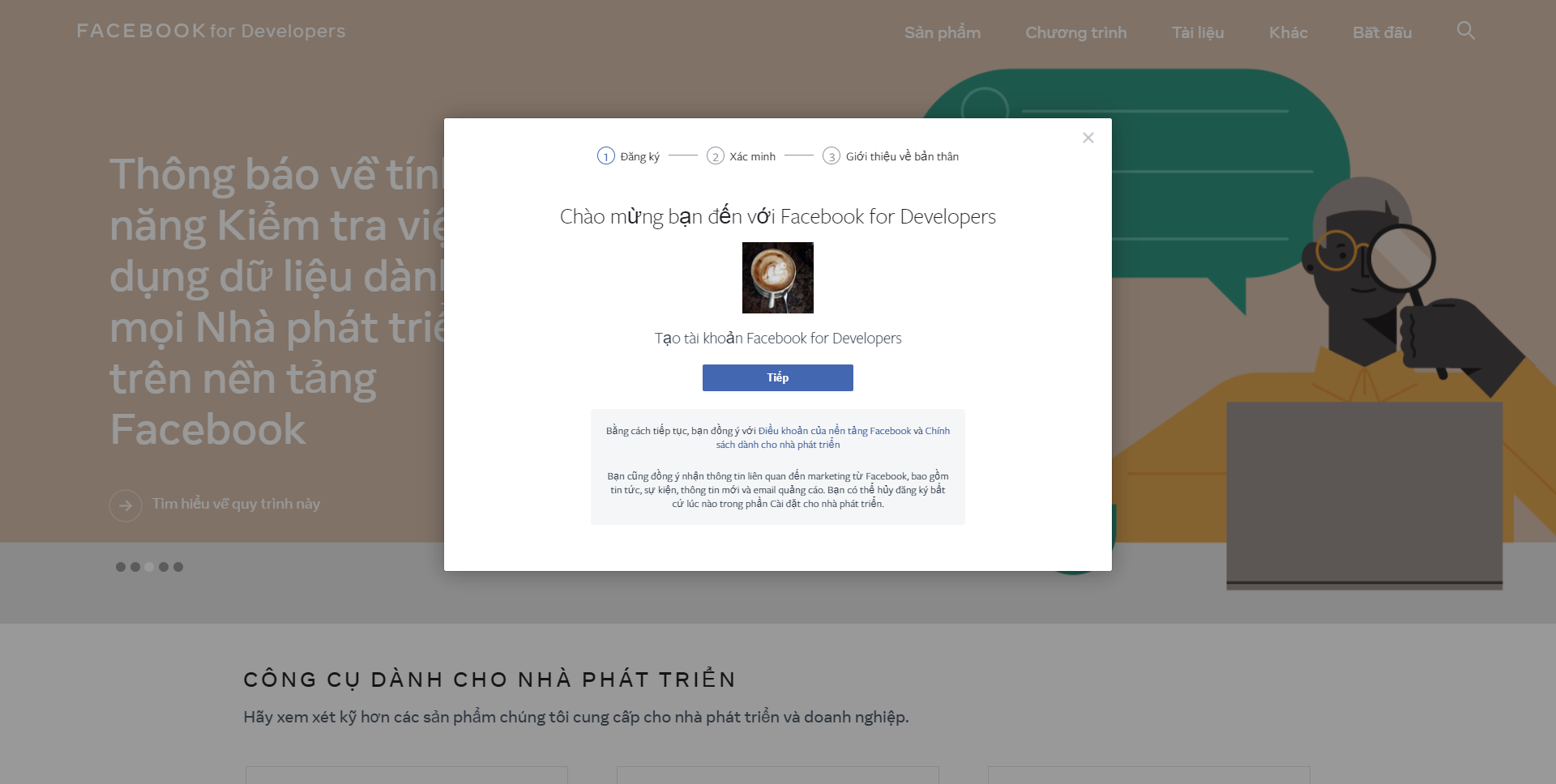
## Cài đặt chương trình

### Đăng ký facebook developer

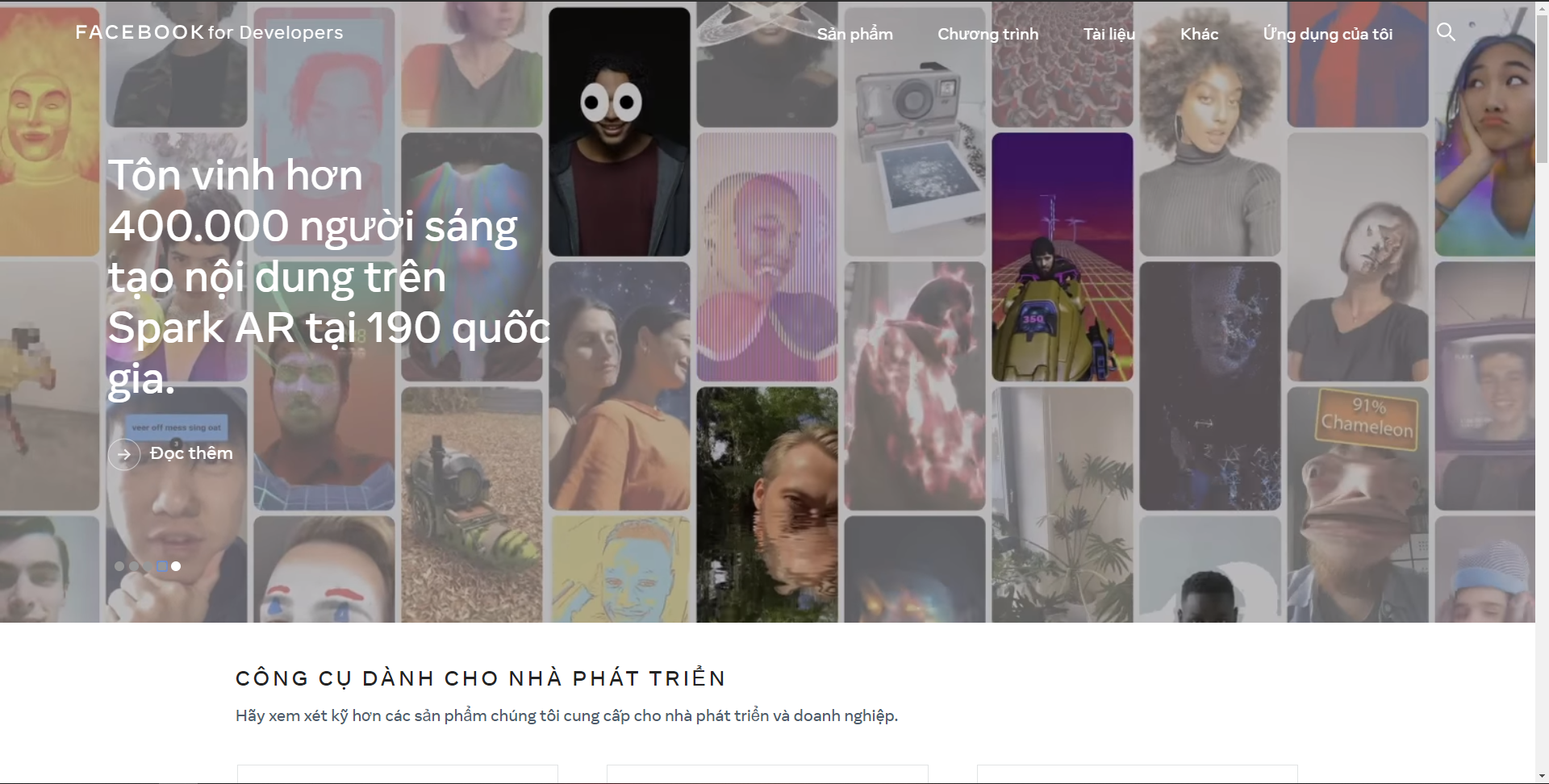
* Truy cập <https://developers.facebook.com>, chọn Đăng nhập



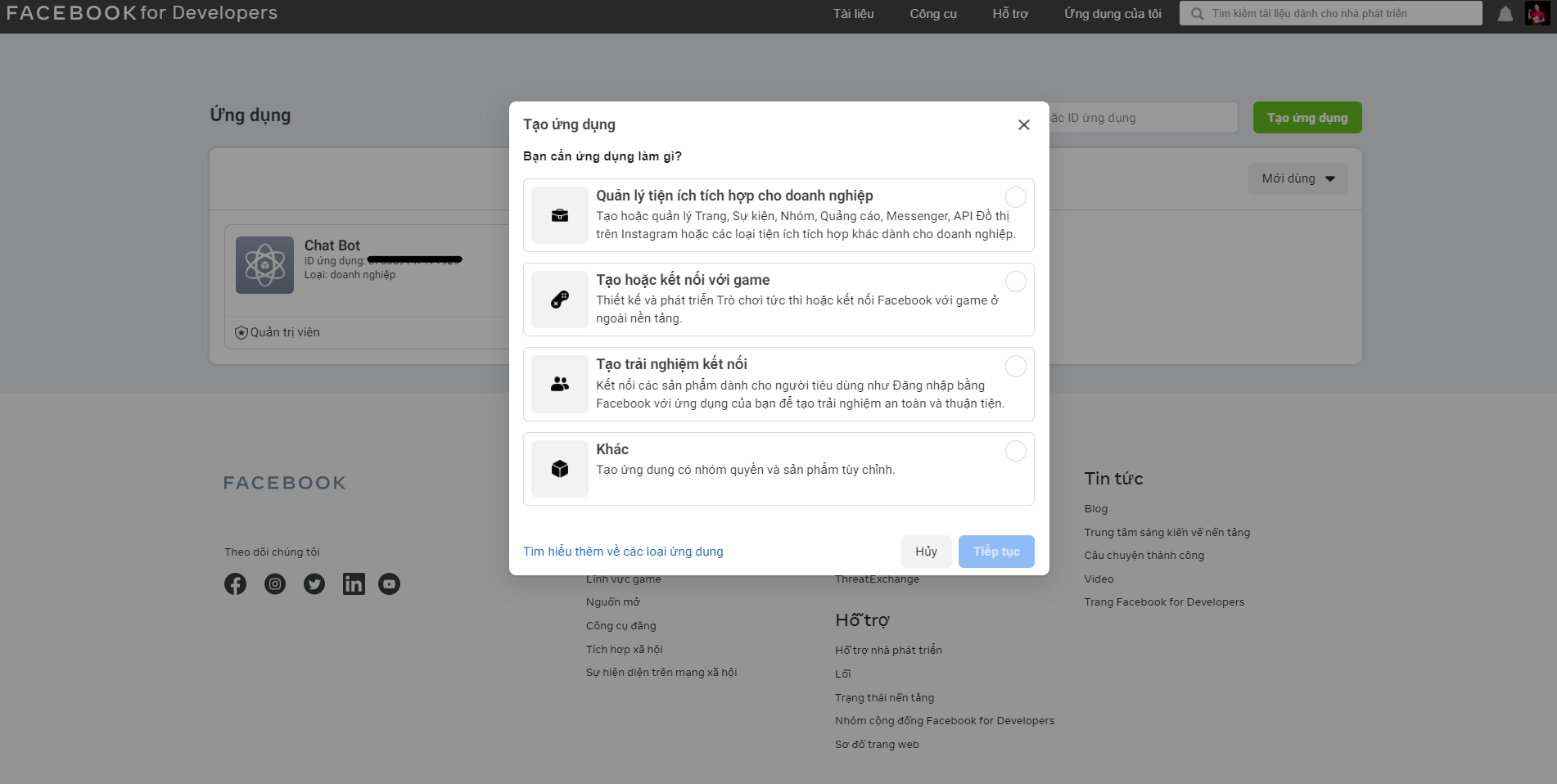
* Lựa chọn Bắt đầu => Tiếp và tiến hành xác minh tài khoản



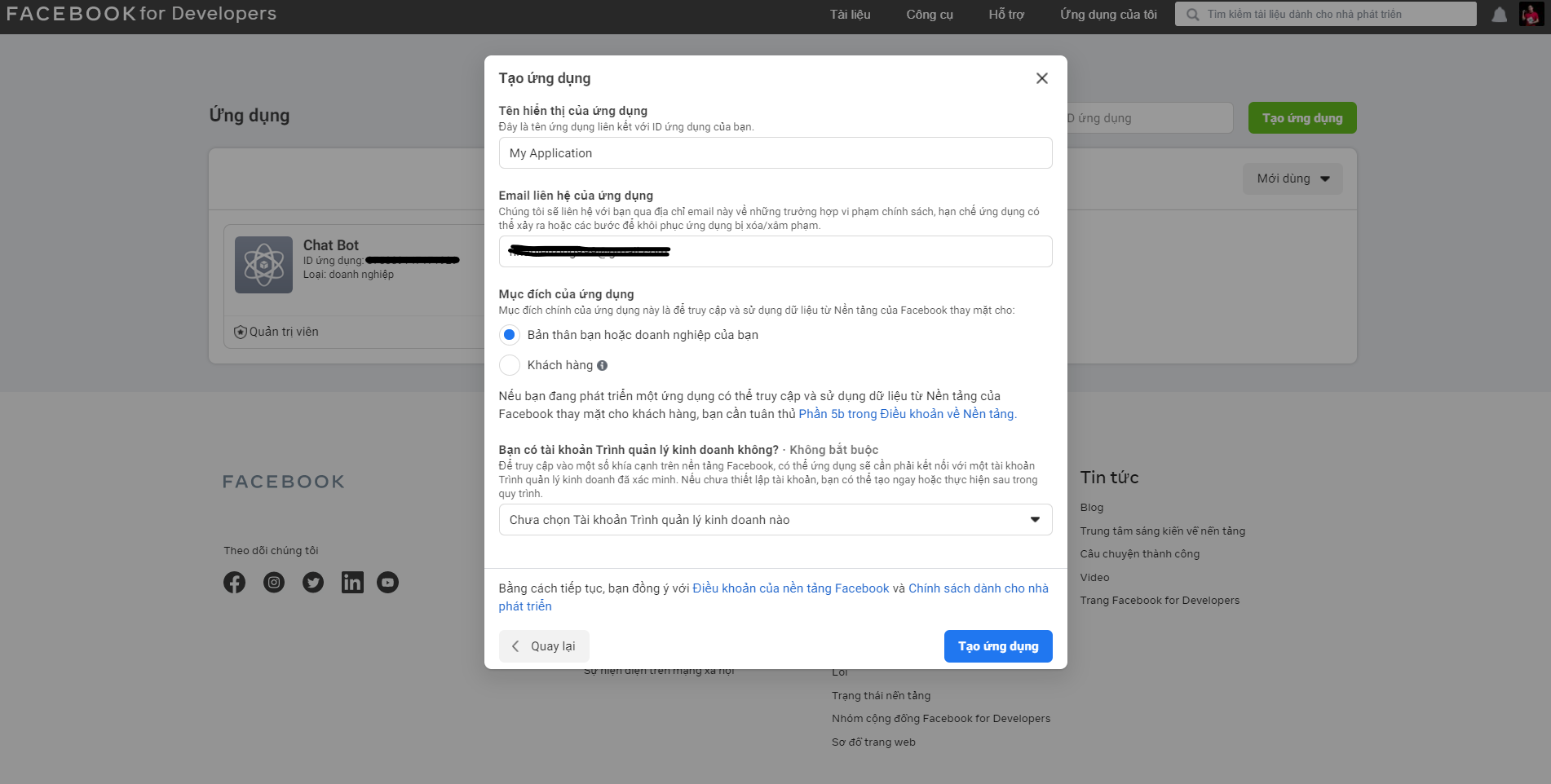
* Sau khi được xác nhận, vào lại, chọn Ứng dụng của tôi



* Chọn Tạo ứng dụng mới => Quản lý tiện ích tích hợp cho doanh nghiệp => Tiếp



* Điền các mục và chọn “Tạo ứng dụng”



* Vào Cài đặt, tạo mã xác thực ứng dụng (PAGE\_ACCESS\_TOKEN):



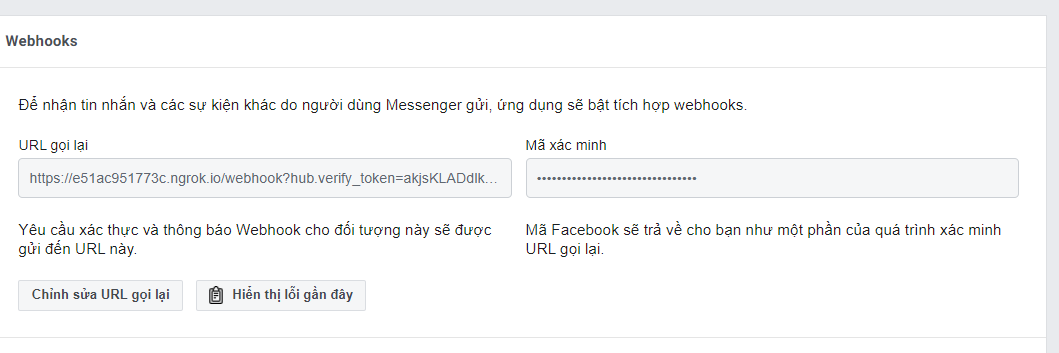


* Cài đặt Webooks:

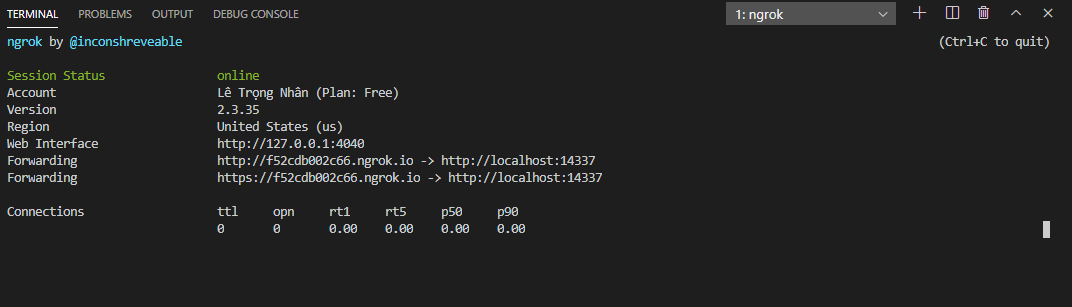
+ Tạo một chương trình cho Webhook-server

|  |
| --- |
| 'use strict';  // Imports dependencies and set up http server  const  express = require('express'),  bodyParser = require('body-parser'),  app = express().use(bodyParser.json()); // creates express http server  // Sets server port and logs message on success  app.listen(process.env.PORT || 1337, () => console.log('webhook is listening'));  // Creates the endpoint for our webhook  app.post('/webhook', (req, res) => {    let body = req.body;  // Checks this is an event from a page subscription  if (body.object === 'page') {  // Iterates over each entry - there may be multiple if batched  body.entry.forEach(function(entry) {  // Gets the message. entry.messaging is an array, but  // will only ever contain one message, so we get index 0  let webhook\_event = entry.messaging[0];  console.log(webhook\_event);  });  // Returns a '200 OK' response to all requests  res.status(200).send('EVENT\_RECEIVED');  } else {  // Returns a '404 Not Found' if event is not from a page subscription  res.sendStatus(404);  }  });  // Adds support for GET requests to our webhook  app.get('/webhook', (req, res) => {  // Your verify token. Should be a random string.  let VERIFY\_TOKEN = "<YOUR\_VERIFY\_TOKEN>"    // Parse the query params  let mode = req.query['hub.mode'];  let token = req.query['hub.verify\_token'];  let challenge = req.query['hub.challenge'];    // Checks if a token and mode is in the query string of the request  if (mode && token) {    // Checks the mode and token sent is correct  if (mode === 'subscribe' && token === VERIFY\_TOKEN) {    // Responds with the challenge token from the request  console.log('WEBHOOK\_VERIFIED');  res.status(200).send(challenge);    } else {  // Responds with '403 Forbidden' if verify tokens do not match  res.sendStatus(403);  }  }  }); |

+ Tạo mã xác minh riêng (một mã tuỳ ý)



+ Sử dụng ngrok để tạo server https trỏ đến server webhook vừa tạo



+ Đưa đường dẫn https vừa tạo được đến mục Chỉnh sửa URL gọi lại (Edit Callback URL):

<ngrok\_https\_server>/webhook?hub.verify\_token=<VERIFY\_TOKEN>&hub.challenge=CHALLENGE\_ACCEPTED&hub.mode=subscribe