

International School

Capstone Project 1

CMU-SE 450

**Installation Guide**

**Version 1.0**

**Date: 25/12/2020**

**viBOTour - Smart Chatbot for Tourist 4.0**

**Submitted by**

**Sang, Nguyen Thanh Thuyen, Pham Van Tuan, Doan Kim**

**Loc, Phung Hoang Phu**

**Approved by Nguyen Duc Man**

#### **PROJECT INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project acronym** | viBOTour | | |
| **Project Title** | Smart Chatbot for Tourist 4.0 | | |
| **Start Date** | 24 Aug 2020 | **End Date** | 5 Dec 2020 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | Man, Nguyen Duc, Ph.D | | |
| **Scrum master**  **/ Project Leader & contact details** | Sang, Nguyen Thanh  Ema[il: thanhsang3111999.ntb@gmail.com](mailto:thanhsang3111999.ntb@gmail.com) Tel: 0344684069  ID: 23211212363 | | |
| **Partner Organization** |  | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
| 2321121321 | Tuan, Doan Kim | [doantuan21101999@gmail.com](mailto:doantuan21101999@gmail.com) | 0913622928 |
| 2321124963 | Loc, Phung Hoang Phu | [phunghoangphuloc@gmail.com](mailto:phunghoangphuloc@gmail.com) | 0912698173 |
| 2321125061 | Thuyen, Pham Van | [phamvanthuyen2802@gmail.com](mailto:phamvanthuyen2802@gmail.com) | 0795697332 |

**REVISION HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Comments** | **Author** | **Approval** |
| 1.0 | 25/12/2020 | Initial Release | All members |  |

# Document Approvals

The following signatures are required for approval of this document.

|  |  |  |
| --- | --- | --- |
| Man, Nguyen Duc  Mentor |  | Date |
| Sang, Nguyen Thanh  Scrum master, DevTeam |  | Date |
| Tuan, Doan Kim  Product Owner, DevTeam |  | Date |
| Thuyen, Pham Van  DevTeam |  | Date |
| Loc, Phung Hoang Phu  DevTeam |  | Date |

TABLE OF CONTENTS

[Document Approvals 1](#_Toc60781933)

[I. Introduction 3](#_Toc60781934)

[II. Detail Installation Guide 3](#_Toc60781939)

[1. Install Node on your Operating System 3](#_Toc60781940)

[2. Install Dialogflow chatbot 3](#_Toc60781941)

[3. Create service API connect to your agent 10](#_Toc60781942)

[4. Integrate Speech chatbot 11](#_Toc60781943)

[5. Setup Backend Node ExpressJS 13](#_Toc60781944)

[6. Run Application 14](#_Toc60781945)

[7. Run Websites 16](#_Toc60781946)

1. **Introduction**

In the era of rapidly developing technology, the concepts of Machine Learning and Artificial Intelligence gradually become popular in all fields. More and more technology ideas and new technologies. With the determination to learn a lot of new technologies, the group has applied many technologies to build applications on both mobile application platforms and websites for management.

The application of a smart chatbot to advise and support tourists in choosing a great tourist destination and a suitable tour will bring a lot of value to both tourists and the national economy. The project team conducts research on chatbot technology with support for both text and voice. The project team uses several available platforms to support this such as Dialogflow, Google Speech to Text.

In addition, to build the application system of 2 web and app technologies, the team has used technologies such as React Native, ReactJS, NodeJS for the backend. The app also uses open libraries and services like Stripe for payments, Open Weather API, and Google Email.

The documentation is designed to help engineers build and maintain the system in the future.

# Detail Installation Guide

## Install Node on your Operating System

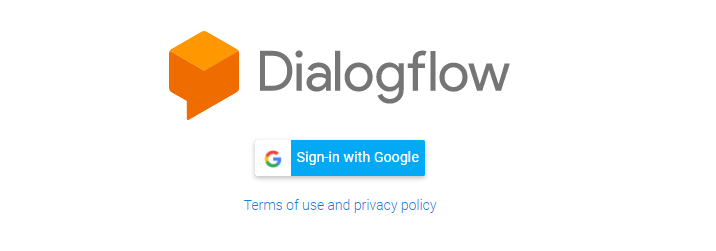
To run React Native, ReactJS and Node Express JS apps in the backend. You must install NodeJS.

Download and install NodeJS to run project. You can follow link: <https://nodejs.org/en/download/>

The stable running version of Node is v14.8.0 and Node Package Manager is v6.14.10

## Install Dialogflow chatbot

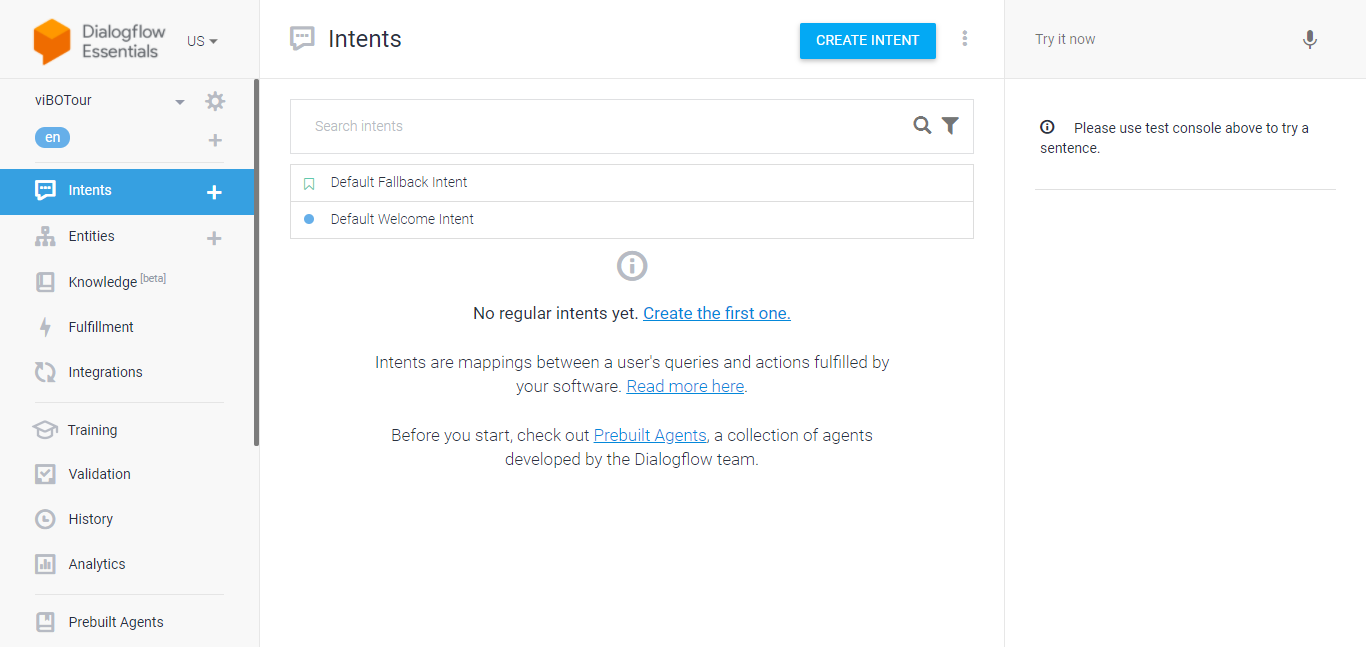
First of all, follow the link: <https://dialogflow.cloud.google.com/> , click Sign-in with google and login to your google account.



Secondly, you can get start if you want to get detailed instructions from Google(You can skip this step and follow my next instructions):

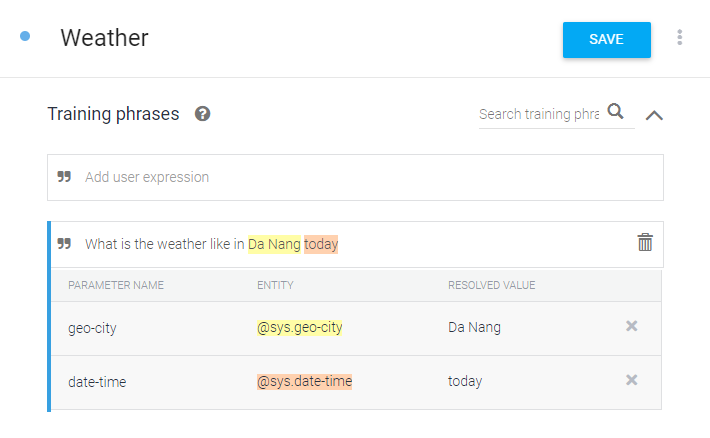
Thirdly, we need to CREATE AGENT.

Enter Enter a name for the agent you want to create. Then, You press button “CREATE” to create new agent. After the system has successfully processed and created the agent, you will receive a page to manage your agent as below:

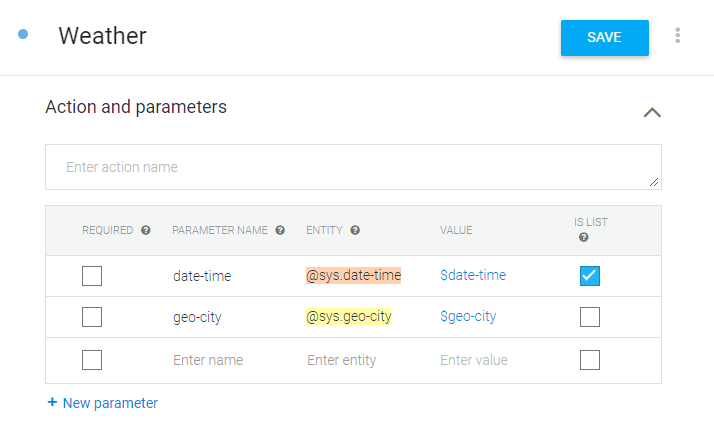


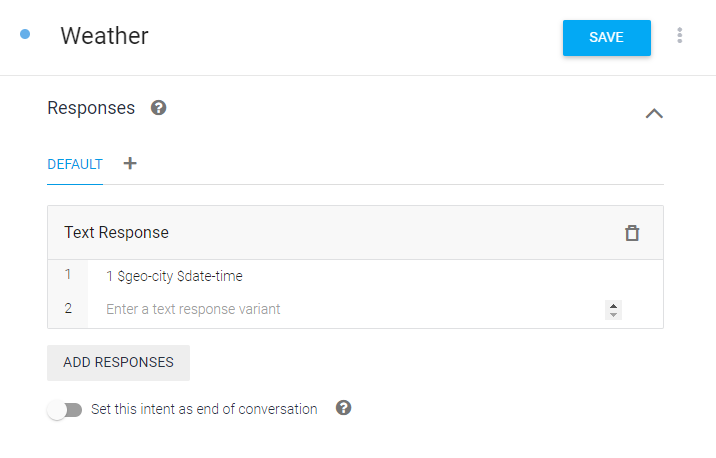
Next, You will be able to use some default intents and entities or create them for your own. Click “CREATE INTENT” to create new intent. Set the name of intent and change some content for it.

Add an event available for the intent you need. In case of inquiring about the weather, it is not needed.



Add training content for chatbot that can understand what you mean. Set the parameters(in entities will be discussed later) to dynamically change the content, but the meaning and the content need to respond to the same.





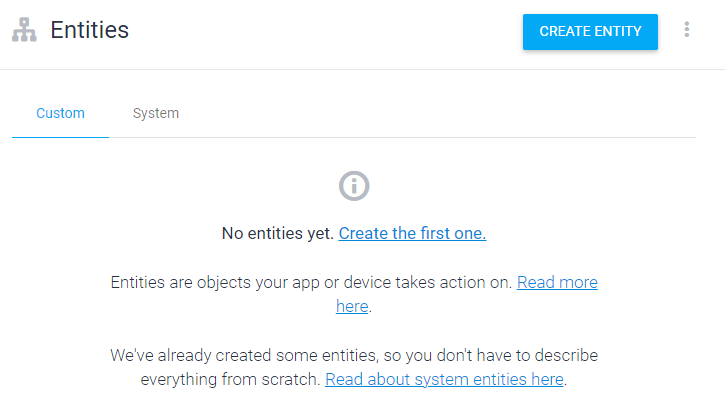
Next is the feedback content. These content chatbot will respond to you. You can use this response for user feedback or handle it in your backend. In the case of asking weather like this. I will get location and time data. We then process it at my backend and then respond the weather information to the user.

These actions and parameters are the list of parameters used in the list of entities. Dialogflow supports some default entities related to city, location, date and time, ... You can also use entities defined by you.

Finally, remember to save and test directly on Dialogflow.

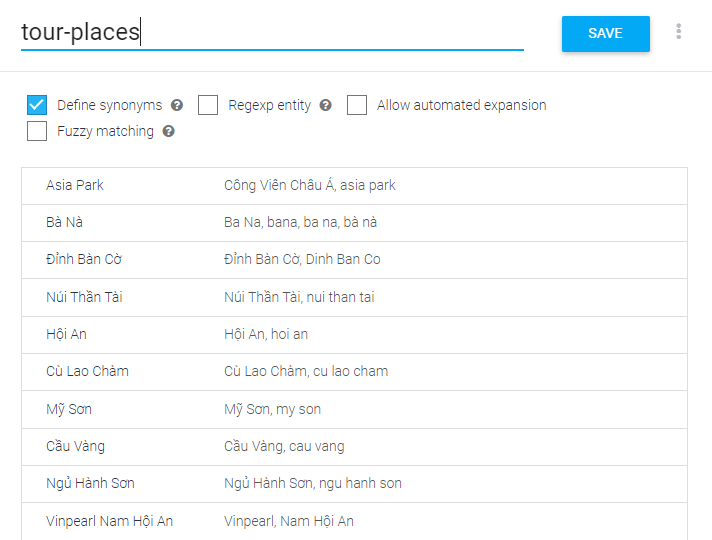
We receive a response(content: ‘1 Da Nang 2021-01-01’) and call the API to OpenWeather to get weather data and give feedback to the user.

Besides intents, you need to use entities to handle the chatbot more flexibly and intelligently.

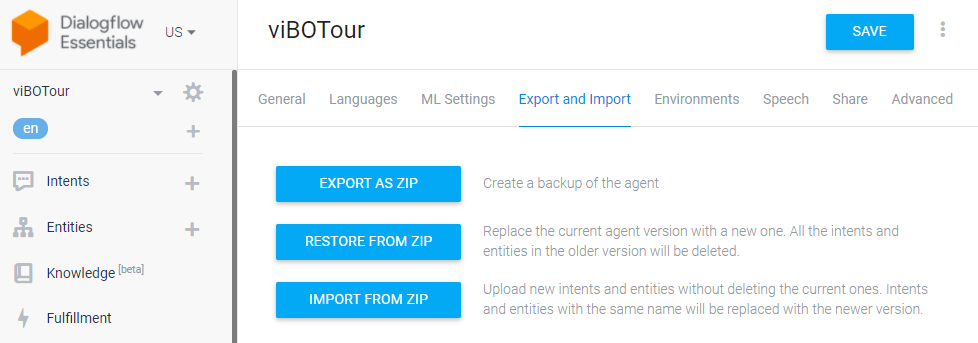


Use available entities or create new it for your own. Click ‘CREATE ENTITY’.

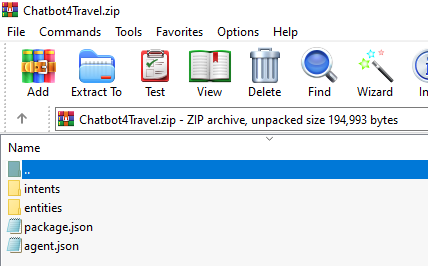
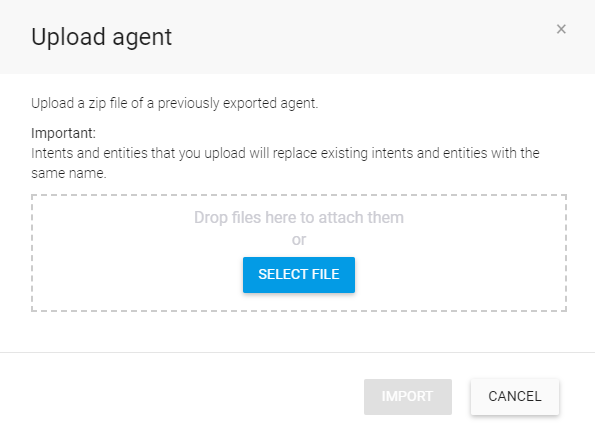
Set name of the entity. Enter the reference value and synonym for it. You will use the synonym to access its value. Don't forget to save it.



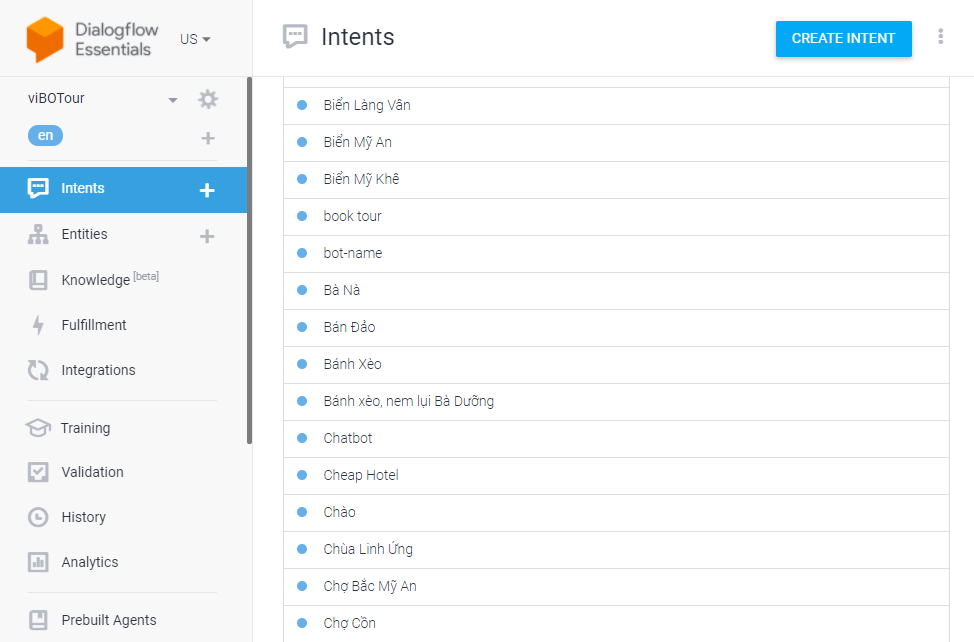
You can import structured intents and entities in setting. Click IMPORT FROM ZIP.



The zip file looks like this. Drop or select file.

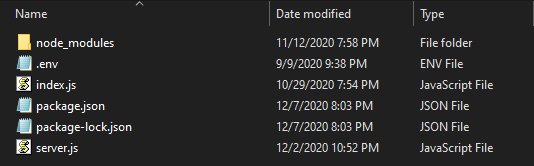
 

Done and set up the intents and entities for the project as follows. These intents and entities are researched and built by us. The project will be continued to expand and develop more in the future.

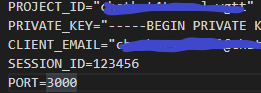


## Create service API connect to your agent

At first, We use NodeJS technology, get start at <https://nodejs.org/en/docs/>. The entire directory after completion will look like this:



Add the environment variable to the .env file:

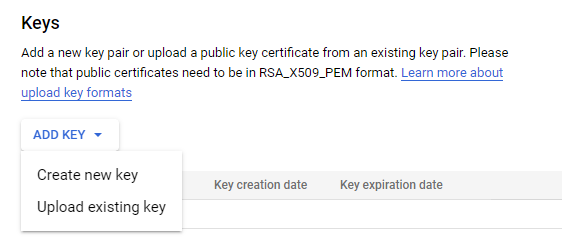


You access to setting of your agent and get Project ID, Private Key and Client Email. Detail at <https://cloud.google.com/dialogflow/es/docs/quick/setup>

You can also follow link to create a service account. Access link <https://console.cloud.google.com/apis/api/dialogflow.googleapis.com/credentials>



You will get a service account in the Credentials section. Click on it to edit. You need to create a key to use it.



Choose JSON type and click CREATE. You will receive a JSON file containing the key you just created, you need to use it to integrate into your project. Carefully manage the JSON file you just received.



Open key file and get your Project ID, Private Key and Client Email. Replace these parameters in the .env file.

Install all packages: npm install

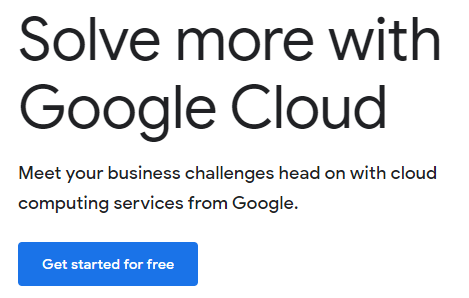
Run server in port 3000 or 5000: npm start

Your API likes: <http://your-ipv4:3000/api/v1/chat> or <http://your-ipv4:5000/api/v1/chat>

## Integrate Speech chatbot

There is a fee for Google's Speech to text service, so you need an international credit card when using it. You can try this service for free for 3 months with a $ 300 free Google customer bonus for the first 3 months.

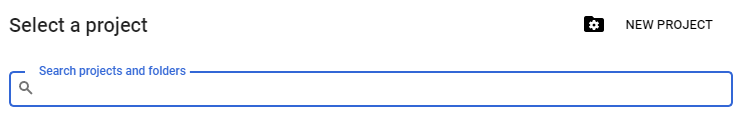
First of all, you follow link [Cloud Computing Services  |  Google Cloud](https://cloud.google.com/) and login to your Google Account.



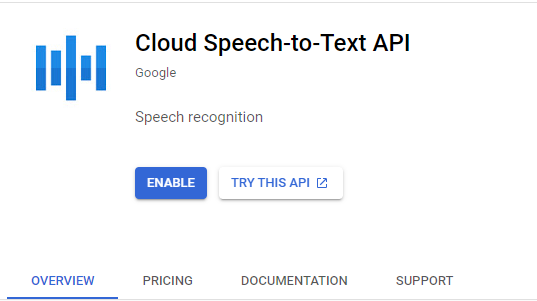
After pressing the button ‘Get started for free’, you need to complete 2 verification steps to use Google Clound. Read the requirements and check the checkboxes if you are familiar with Google's policies.

Continue setting up your billing information.

After a successful setup, you will receive a $ 300 donation for the first 3 months. Next you need to create a project in order to access the services. Remember to check this project for the settings related to the service you use.



Start searching and using Google's Speech to Text service.



Read carefully the information items of this service, then proceed to set the payment and enable this service. Follow link: <https://console.developers.google.com/apis/credentials> . Choose your project and create a Credentials.

Click on Service account. Complete 3 steps to create a service account.

You will get a service account in the Credentials section.

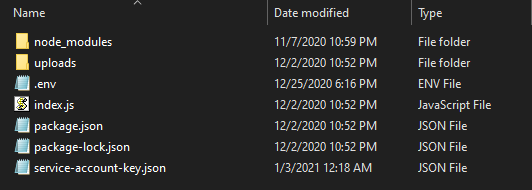
Click on it to edit. You need to create a key to use it. Click on “Create new key”.

Choose JSON type and click CREATE. You will receive a JSON file containing the key you just created, you need to use it to integrate into your project. Carefully manage the JSON file you just received.



The next step is to build the code to use the key to connect to Google's Speech to Text service. You need to send an audio file to Google to separate and filter the text and respond to the results for you.

Place the JSON file containing the received Service account key in the same directory as the backend.



Create an environment variable stored in the .env file as follows:

URL\_IP=0.0.0.0

CREDENTIAL=service-account-key.json

You also need to use ffmpeg to get the encoding, hZ, etc for an audio file. Detail at <https://ffmpeg.org/documentation.html> . Read this document and install it on your server Operating System

Install node packages: npm install

Run it: npm start

All done, you can use your API to send audio to Google Speech to Text and implement integration into chatbot. API Post: <http://your-ipv4:3005/speech>.

## Setup Backend Node ExpressJS

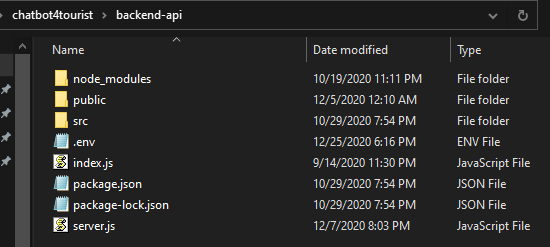
First of all, Install Postgresql and pgAdmin4.0 on your machine. Detail at <https://www.pgadmin.org/>

Secondly, Create new database, setup tables or restone from vibotour\_backup\_final.sql.

Thirdly, Set up Stripe account for payment and get the public key and secret key for integration. See details at <https://stripe.com/docs/>

Next, Use Google mail service. See instructions at <https://blog.mailtrap.io/nodemailer-gmail/>

The backend directory is as follows:



Note CHANGING SOME of the corresponding fields below in the .env file

DATABASE\_URL=postgresql://user-postgres:password@localhost:port/dbname

EMAIL = your-google-mail

PASSWORD = password-google-mail

STRIPE\_SECRET\_KEY=sk\_test\_stripe

STRIPE\_PUBLIC\_KEY=pk\_test\_stripe

SERVER\_IP=http://your-ip-v4:3001

Install packages: npm install

Run web: npm start

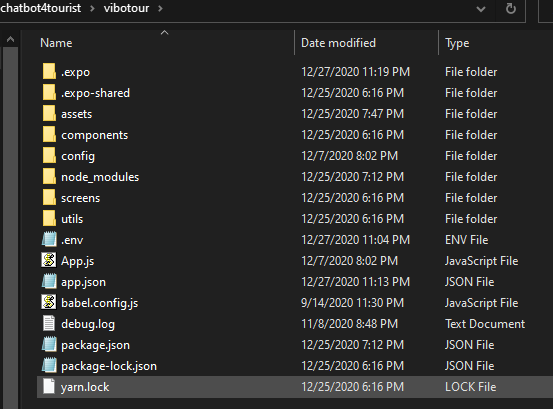
## Run Application

You need to learn how to use Expo. See technology documentation at <https://docs.expo.io/>

Get the client id information for login with google at <https://docs.expo.io/versions/latest/sdk/google/>

Integrating weather for chatbot need to use OpenWeatherMap API. Read and get Key at <https://openweathermap.org/api> .

The folder likes:



Next, You change environment variables in .env file.

DATABASE\_URL=postgresql://user-postgres:password@localhost:port/dbname

SERVER\_IP=http://your-ipv4:3001

CHATBOT\_SERVER\_URL=http://your-ipv4:3000/api/v1/chat

SPEECH\_SERVER\_URL=http://your-ipv4:3005/speech

WEATHER\_kEY=Open-weather-API-key

androidClientId = expo-android-client-id

iosClientId = expo-ios-client-id

androidStandaloneAppClientId = android-client-id

iosStandaloneAppClientId = ios-client-id

STRIPE\_PUBLIC\_KEY=pk\_test\_your-stripe-public-key

Install packages: npm install

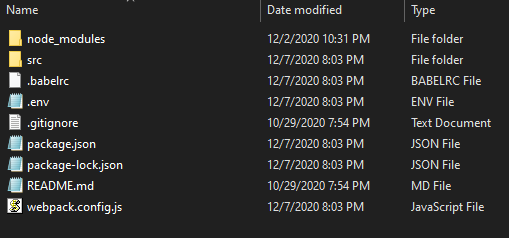
Run web: npm start

You can test the app directly by installing the Expo app at the App Store or Google Play.

With Android operating system, you can download and use our app on Google Play at <https://play.google.com/store/apps/details?id=com.teamse007.vibotour>

## Run Websites

The directory containing the source code of the Frontend website is as follows:



Change the information corresponding to your backend in the .env file. Switch to the IP server corresponding to your machine in this file.

Install packages: npm install

Run web: npm start