Overview

This documentation covers Whoxa Web Setup, which sold through CodeCanyon.

Whoxa chat

First of all, Thank you so much for purchasing this item and for being our loyal customer.

You are awesome!

With your purchase, you are entitled to get free lifetime updates to this product.

- Whoxa chat is a complete whatsapp like web chat solution designed for operating various industries to manage employees or users communication from one platform. Developed for convenience, whoxachat offers a seamless experience through desktop or laptop web interfaces to support any browsers.
- It is built using ReactJS as frontend and NodeJS as backend framework to offer a robust and scalable solution.
- React.js, commonly referred to as React, is a popular JavaScript library developed by Facebook for building user interfaces, especially single-page applications. React enables developers to create reusable UI components and efficiently update and render the right components when data changes, making it widely used for dynamic and interactive web applications.
- Node.js is an open-source, cross-platform runtime environment built on Chrome's V8 JavaScript engine, designed for executing JavaScript code on the server side. Traditionally, JavaScript was only used for front-end development in web browsers, but Node.js has extended JavaScript's capabilities to back-end and full-stack development, enabling JavaScript to handle server-side logic and data handling.
- PostgreSQL is a powerful, open-source **relational database management system (RDBMS)** used for storing and managing data. It is one of the most advanced SQL databases in the world.

Whoxa chat consists of the following libraries and third party APIs:

Socket.IO :

A Socket.IO chat application enables real-time communication between users by establishing a persistent connection between the server and clients. This type of setup is widely used for chat applications, as it allows instant message delivery and responsiveness

Twilio:

Twilio is cloud communications platforms, Twilio Provides global SMS otp for phone number verification or authentication.

Getting Started

- First, you Download the source code first from CodeCanyon.
- Right now, we are giving documentation for installing the whoxa chat system on VPS servers. If you want to setup on Cpanel then you need to connect here on whatsapp or email on primocys@gmail.com.
- Whoxa chat have complete one click easy installation solution for as you don't need to run any specific command to install nodejs, npm or yarn.

If you don't know how to download the source code follow the following steps:

- Log into your Envato Market account.
- Hover the mouse over your username at the top of the screen.
- Click Downloads from the drop-down menu.
- Click All files and documentation.

What you'll need

- **VPS :**
  - First you have to buy VPS from hosting provider like Hostinger, AWS ec2, etc.
  - You have to buy ubuntu server.
- **Root User Permissions :**
  - You will need sudo user permission on the server to automate the project setup.

Setup ssh

- **SSH :**
  - You can use ssh password or you can generate ssh key

    - You have to generate public and private key and upload public key on the server by hpanle if your vps is from hostinger
    - If your vps is ec2 then you already added the public key during the launch of instance.
- **SSH into server :**
    - Then after you have to connect to server using SSH

Whoxa Installation Guide (Linux Live Environment)

This document provides a complete step-by-step guide to set up and deploy Whoxa on a Linux local environment. It covers MySQL, Node.js, project deployment, PeerJS setup, and auto-run configuration.

If you want to run on Linux (Local System) then follow below link:
https://document.whoxachat.com/docs/Linux%20Local%20System/Intro

Getting Started

These are the steps if you want to setup Whoxa manually.

Steps

- PostgreSQL Installation
- Node.js Installation
- Project Deployment
- PeerJS Server Setup
- Auto Run Project on Server Restart

PostgreSQL Setup

1. Run this Command in Linux Terminal to install PostgreSQL:
sudo apt install postgresql postgresql-contrib

2. Start and Enable PostgreSQL, You can run following commands to start and enabale PostgreSQL:
sudo systemctl start postgresql
sudo systemctl enable postgresql

3. Password Protection:

3.1 Switch to the PostgreSQL User

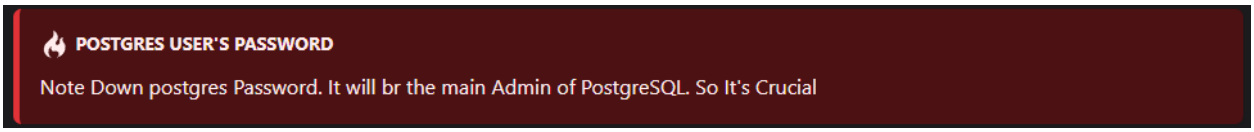
sudo -i -u postgres

3.2 Open PostgreSQL terminal

psql

3.3 You can now set password

ALTER USER postgres WITH PASSWORD 'Root@1142';



4. Close the PostgreSQL terminal

\q

5. Exit from posgres user

exit

Allow Password Authentication

1. Edit the pg_hba.conf file:

sudo nano /etc/postgresql/*/main/pg_hba.conf

2. Find this line:

source ~/.bashrc

3. change this line

local all postgres md5

4. Install latest Node version:

nvm install --lts

5. Verify Version:

node -v

Restart PostgreSQL

sudo systemctl restart postgresql

Database Creation

To Login

```
psql -U postgres -h localhost
```

Create Database

```
CREATE DATABASE mydatabase;
```

Verify Database

```
\l
```

Create Users

Create Users

```
CREATE USER myuser WITH PASSWORD 'mypassword';
```

List users

```
\du
```

Give Previlidges

Give full access

GRANT ALL PRIVILEGES ON DATABASE mydatabase TO myuser;

Deployment

1. Extract whoxa.zip → You will get three zip files:
 - whoxa_admin.zip
 - whoxa_frontend.zip
 - whoxa_backend.zip

2 Upload whoxa-backend.zip to a location where you want to deploy Project using FileZilla

/var/www/whoxa (This is the recommended location)

3. Install unzip to unzip the Project . Run following command to install unzip

sudo apt install unzip

unzip whoxa-backend.zip

ENV File Configurations

Make a .env file from the demo.env in the project on folder which contain package.json with following content.

Port="3047"
NODE_ENV="production"
screteKey="screteKey"
DB_USERNAME="postgres"
DB_PASSWORD="root@1142"
DB_DATABASE="whoxa_advanced"
JWT_EXPIRY="24h"
baseUrl="http:{{HOST}}:{{PORT}}"
TWILIO_ACCOUNT_SID="TWILIO_ACCOUNT_SID"
TWILIO_AUTH_TOKEN="TWILIO_AUTH_TOKEN"
TWILIO_PHONE_NUMBER="TWILIO_PHONE_NUMBER"
MSG91_SENDER_ID="MSG91_SENDER_ID"
MSG91_API_KEY="MSG91_API_KEY"
MSG91_TEMPLATE_ID="MSG91_TEMPLATE_ID"
IS_TWILIO_ENABLED=true

IS_MSG91_ENABLED=false
IS_CLIENT=true

ONE_SIGNAL_APP_ID="ONE_SIGNAL_APP_ID"
ONE_SIGNAL_API_KEY="ONE_SIGNAL_API_KEY"
ANDROID_CHANNEL_ID="ANDROID_CHANNEL_ID"

Server Starting

1. Open terminal at the project root (where package.json is located).

2. Run the following command to install dependencies:
npm install

3. After successful running of npm install install global dependency to auto configure the DB related tasks by running following command.
npm install -g sequelize sequelize-cli

4. To install PM2 Globally
npm install -g pm2

4. To run server run following command within the terminal where package.json is located:
pm2 start index.js --name whoxa

PeerJS Installation

1. Install PeerJS globally:
npm install -g peer

2. Run following command to run peer server on port 4001:
pm2 start "peerjs --port 4001" --name "peerjs-server"

Auto restart on server reboot

1. Run following command to save all the process running on pm2

pm2 save

2. Run following command to restart all the saved processes on server restart

pm2 startup system

3. Run following command to see running pm2 processes

pm2 status

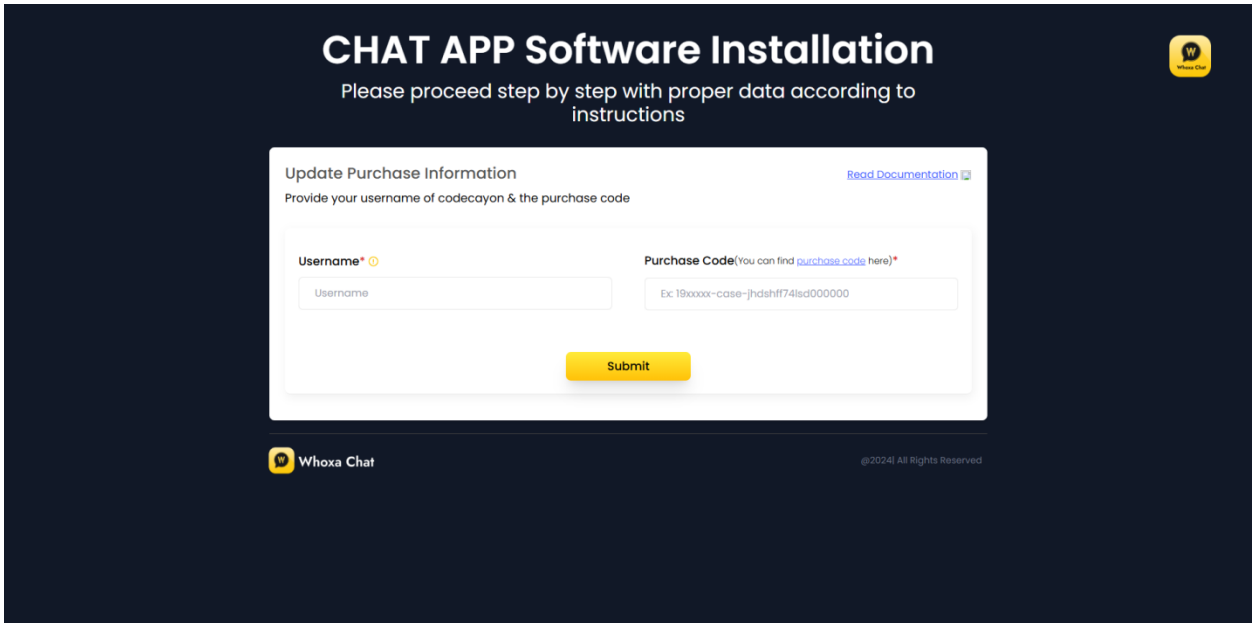
4. Run following command to see logs of a process

pm2 log {index}

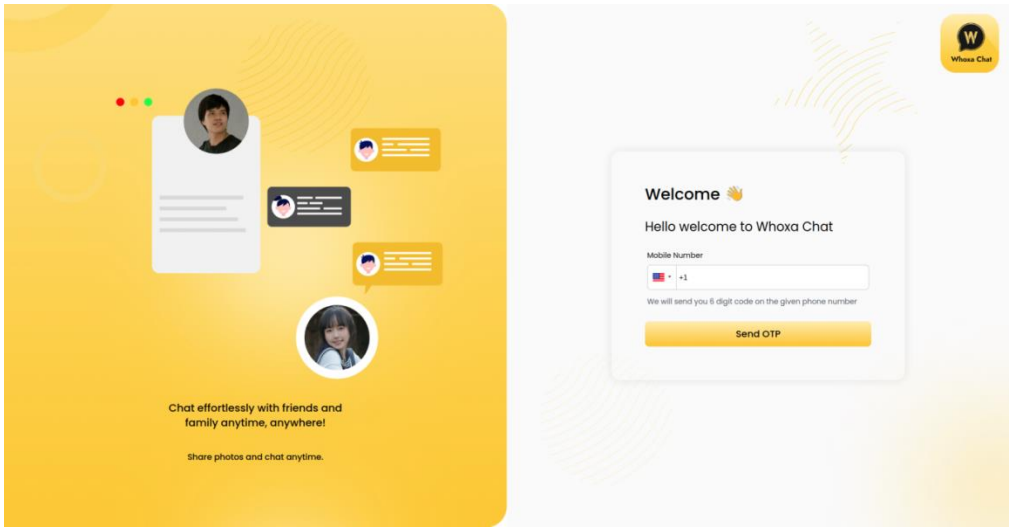
Accessing the App

Search following URL to see Frontend:
http://{your_server_ip}:{port_on_which_server_is_running}

First it will show purchase code verification page



After purchase code verification done then it will show this page:



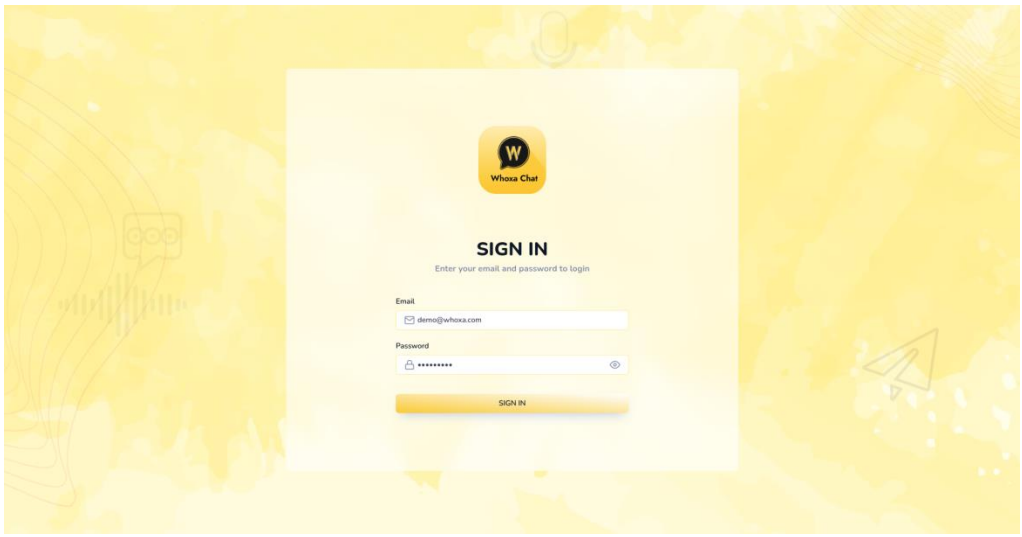
Search following URL to see Admin Panel
http://{your_server_ip}:{port_on_which_server_is_running}/admin

Search following URL to see Website(Frontend)
http://{your_server_ip}:{port_on_which_server_is_running}

Search following URL to see Api Endpoint
http://{your_server_ip}:{port_on_which_server_is_running}/api

Login Details:

demo@whoxa.com
Admin@123



Whoxa – Windows (Local System) Installation Guide

Getting Started

These are the steps if you want to set up Whoxa manually on a Windows (Local System).

Steps

- MySQL Installation
- Node.js Installation
- Project Deployment
- PeerJS Server Setup
- Auto Run Project on Server Restart

MySQL Setup (XAMPP)

Installation of XAMPP

1. Download XAMPP for Windows from the official website:
https://www.apachefriends.org/download.html

2. Run the installer and select the following components during installation:
- Apache
- MySQL
- PHP
- phpMyAdmin (recommended)

3. Once installation is complete, open the XAMPP Control Panel.

Starting MySQL Server

In the XAMPP Control Panel, click Start next to MySQL.

Tip: You can also start Apache if you plan to use PHP or phpMyAdmin.

When MySQL is running, its status will turn green.

Accessing MySQL

1. Open phpMyAdmin by clicking the Admin button next to MySQL in XAMPP Control Panel, or by visiting http://localhost/phpmyadmin
2. By default in XAMPP, the root user has no password.
3. Create database through phpMyAdmin named 'whoxa'.

Node Setup (Without NVM)

Installation of Node.js

1. Download the Node.js Installer for Windows from: https://nodejs.org/en/download (You can download latest version.)
2. Choose the LTS (Long-Term Support) version for stability.
3. Run the installer and follow setup wizard:
   - Accept license agreement
   - Choose installation folder
   - Select default options
   - Install Node.js and npm (Node Package Manager)
4. Restart terminal or Command Prompt after installation.

Verify Node.js Installation

Run the following command:

node -v

Project Extraction

1. Extract whoxa.zip → You will get three zip files:
   - whoxa_admin.zip
   - whoxa_frontend.zip
   - whoxa_backend.zip
2. Extract whoxa_backend.zip → You will get:
   - automate.sh
   - auto_deploy.zip
3. Extract auto_deploy.zip → You will get automation files + whoxa.zip
4. Extract whoxa.zip to desired location, e.g., C:/whoxa or D:/whoxa.

Config File

Make a config.json inside config folder of Project And add following content as per you MySQL and Server Configuration:

```
{
  "development": {
```

```
  "username": "root",
  "password": "",
  "database": "whoxa",
  "host": "127.0.0.1",
  "dialect": "mysql"
 },
 "test": {
  "username": "root",
  "password": null,
  "database": "database_test",
  "host": "127.0.0.1",
  "dialect": "mysql"
 },
 "production": {
  "username": "root",
  "password": null,
  "database": "database_production",
  "host": "127.0.0.1",
  "dialect": "mysql"
 }
}
```

ENV File Configurations

Create a .env file in the project folder (where package.json is located) with the following content(If you already have .env file then only change below values other values will be same):

```
TWILIO_AUTH_TOKEN=""
TWILIO_FROM_NUMBER=""
TWILIO_ACCOUNT_SID=""
baseUrl="http://localhost:3000/"
```

Server Starting

1. Open terminal at the project root (where package.json is located).
2. Run the following command to install dependencies:

```
npm install
```

3. Install global dependency for DB tasks:

```
npm install -g sequelize sequelize-cli
```

4. Run server with:
npm run dev

PeerJS Installation

1. Install PeerJS globally:

npm install -g peer

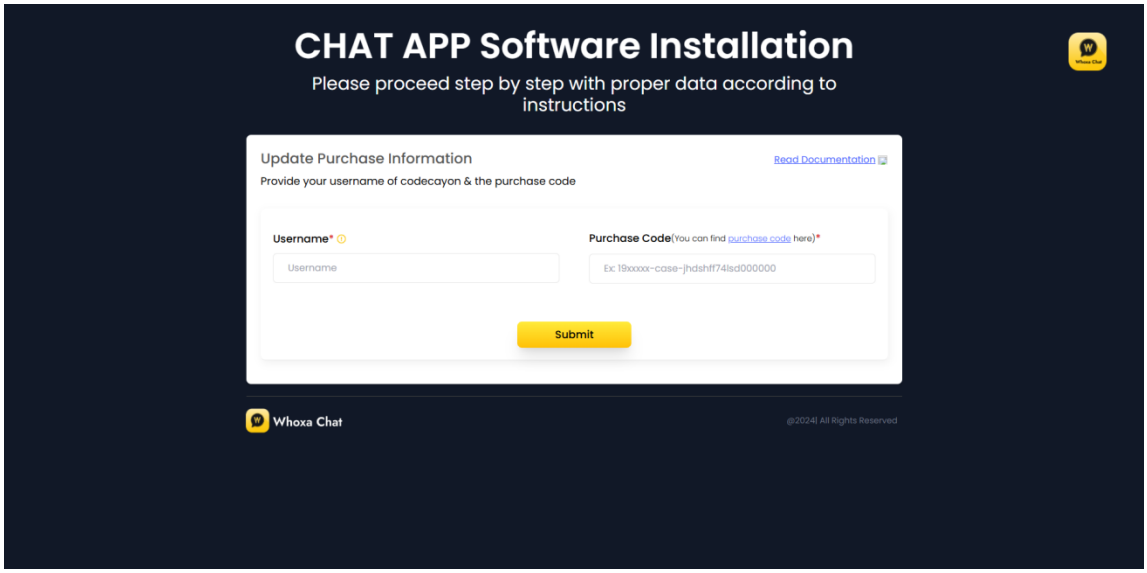
2. Run PeerJS server on port 4001:

peerjs --port 4001

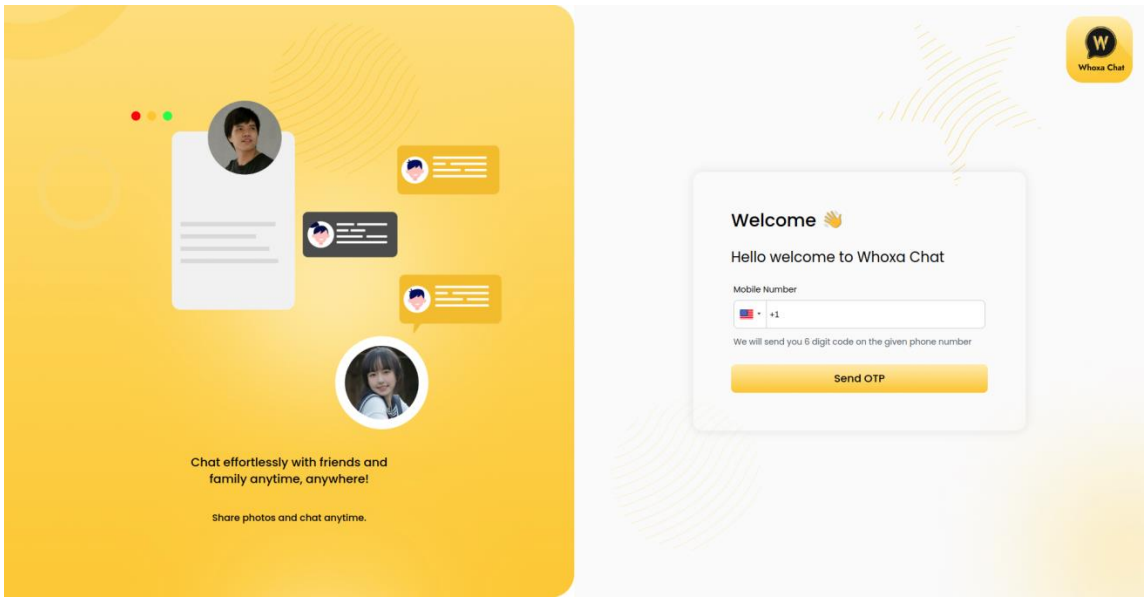
Accessing the App

Search following URL to see Frontend:
http://{your_server_ip}:{port_on_which_server_is_running}/

First it will show purchase code verification page



After purchase code verification done then it will show this page:



Search following URL to see Admin Panel
http://{your_server_ip}:{port_on_which_server_is_running}/admin

Login Details:
demo@whoxa.com
Admin@123