GROUP V

PRODUCT DOCUMENTATION

Ronald Chipoco and Von Miones

System Requirements

The Local Government Unit Health Surveillance System is designed to run on Windows or Linux operating systems. The minimum system requirements for running the system are:

- Windows 7 or later, or
- Linux operating system
- Core i3 Processor or equivalent
- 4GB of RAM
- 512GB of hard drive storage for Windows PCs
- 8 port network switch
- Network router
- At least 5mbps internet connection for publishing remote IP for remote demonstration

The Local Government Unit Health Surveillance System is compatible with both 32-bit and 64-bit versions of Windows and Linux. It is recommended that users have the latest service packs and updates installed for their operating system to ensure the system runs smoothly and without any issues.

In addition to the minimum system requirements, the Local Government Unit Health Surveillance System requires a web browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge, to access the user interface.

Overall, the system requirements for the Local Government Unit Health Surveillance System are designed to ensure that the system can be easily deployed and used by local government units of all sizes and resource levels, regardless of their choice of operating system.

Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

Software requirements

The system requires the following software:

- WinNMP - Nginx MariaDB MongoDB Redis Php 8 development stack for Windows

Features

- Easy to upgrade! Backups, configuration files, database data, included libraries and projects are persistent during upgrades (not overwritten)
- A single installer for both x86 and x64 systems, with 32bit and 64bit versions of MariaDB, PHP and WinNMP Manager
- Lightning-fast web server. Optimized for best performance
- Tools: mysql client, php console, composer, acmePhp, hostsEditor, putty, mailToDisk
- PORTABLE: you can move it to a different location, configuration files are updated automatically
- For easy access, all configuration files are stored in one place: WinNMP\conf and all log files are stored in one folder: WinNMP\log
- Multiple PHP Versions and version switcher. Dynamic number of php-cgi processes
- The server manager runs minimized in the system tray and monitors, logs and restarts crashed servers, just like php-fpm on Linux
- Project setup, Local Virtual Servers for projects, Upload, Sync and Browse with WinSCP
- Optional database daily backups can be enabled with --backup. Up to 7 backup files will be created per database per weekday

Current Package Contains

- Nginx 1.21.3 web server
- MariaDB 10.4.21 database server, mysql 5.5.5 replacement (32/64bit)
- MongoDB 4.2.17 document-based database (64bit)
- Redis 5.0 Cache/NoSql, memcached alternative (64bit)
- Php 5.6.40 & PHP 7.2.34 & PHP 7.3.31 & PHP 7.4.24 & PHP 8.0.11 scripting language (32/64bit)
- XDebug, GeoIP, Gender, Mongodb PHP Extensions
- WinSCP SFTP client
- HTTPS using free LetsEncrypt certificates
- Composer dependency manager for php
- WP-CLI command-line interface for WordPress
- Adminer web-based database manager
- Reg.php regular expressions tester
- WinNMP Manager (32/64bit), formerly known as WTServer

Contact Us :

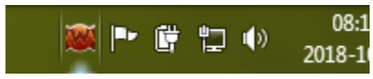
Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

Installing and Upgrading:

- DOWNLOAD AND EXECUTE THE LATEST INSTALLER. The installer produces a portable folder.
- The same installer can be re-run later to download and update your stack with the latest MariaDB, Mongo or PHP version.
- Any previous projects, settings, databases and PHP extensions will NOT be lost or deleted during upgrades/updates.
- It is safe to choose any destination folder as long as it is publicly accessible, if you don't like C:\WinNMP

Getting Started:

- Start WinNMP
- Look in the taskbar for the WinNMP icon: 
- Press Options and check Start WinNMP with Windows
- Type in your browser http://localhost
- Read WINNMP BASIC USAGE - GETTING STARTED
- The default MariaDb(MySql) host is localhost with username root and no password.
- The default Redis host is localhost port 6379

System Initialization

Manual Database Creation

Once all the requirements have been met and the server has been set up, follow these steps:

1. Create a database using the manual MySQL database creation method:
   - Open MySQL console
   - Enter the following command to create a blank database named dict_tot:

     ```
     CREATE DATABASE dict_tot;
     ```

2. Table Creation: Diseases

   Use the following SQL script to create a table named "diseases" with the following fields:

   SQL Script:

Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

```
CREATE TABLE diseases ( id INT(11) NOT NULL, name VARCHAR(255) NOT NULL, description TEXT NULL, symptoms TEXT NULL, treatment TEXT NULL, dtcreated DATETIME NOT NULL, dtupdated DATETIME NULL, datastatus ENUM('Active', 'Inactive') NOT NULL, PRIMARY KEY (id) );
```

3. Table Creation: Demoentity

   Use the following SQL script to create a table named "demoentity" with the following fields:

   SQL Script:

```
CREATE TABLE demoentity ( id INT(11) NOT NULL AUTO_INCREMENT, fname VARCHAR(50) NOT NULL, mname VARCHAR(50), lname VARCHAR(50) NOT NULL, suffix VARCHAR(10), sex VARCHAR(10) NOT NULL, civilstatus VARCHAR(20) NOT NULL, birthdate DATE NOT NULL, height FLOAT(4, 1), weight FLOAT(4, 1), bloodtype VARCHAR(5), barangay VARCHAR(50) NOT NULL, citymun VARCHAR(50) NOT NULL, province VARCHAR(50) NOT NULL, email VARCHAR(100), contactno VARCHAR(20) NOT NULL, contactno2 VARCHAR(20), isvaccinated ENUM('Yes', 'No') NOT NULL DEFAULT 'No', vaccinedetails VARCHAR(255), disease VARCHAR(50), symptoms TEXT, medicationdetails TEXT, recommendation TEXT, dtcreated DATETIME DEFAULT CURRENT_TIMESTAMP, dtupdate DATETIME ON UPDATE CURRENT_TIMESTAMP, remarks TEXT, datastatus ENUM('Active', 'Inactive') NOT NULL DEFAULT 'Active', temp FLOAT(4, 1), coviddiagnosed ENUM('Yes', 'No') NOT NULL DEFAULT 'No', covidencounter ENUM('Yes', 'No') NOT NULL DEFAULT 'No', country VARCHAR(50) DEFAULT 'Philippines', PRIMARY KEY (id) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Contact Us :

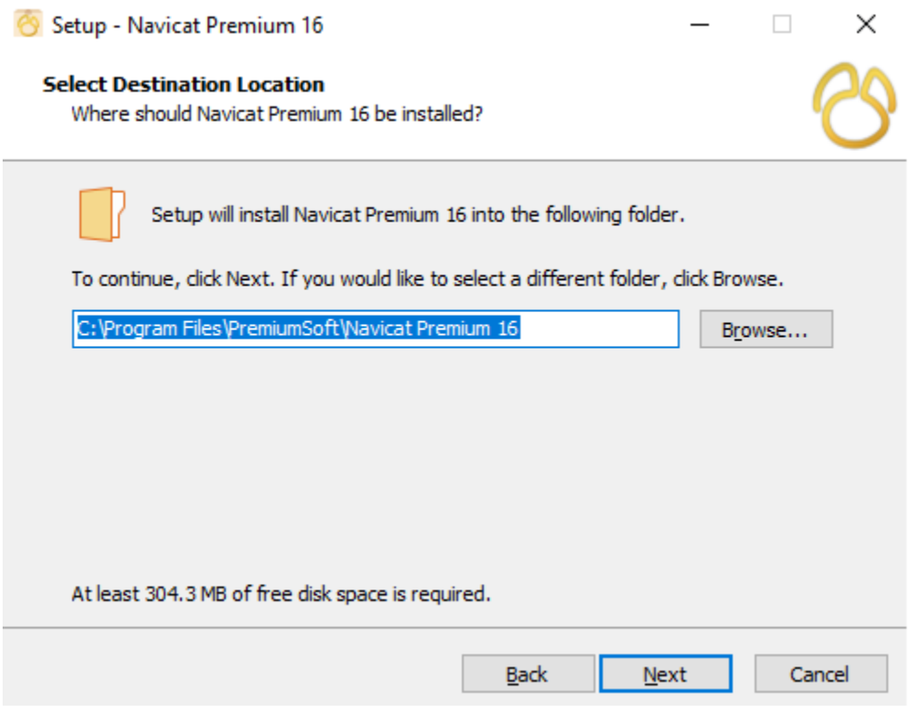
Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

Creating Database and Table with Navicat

Installing Navicat

First, we need to download and install Navicat. Navicat is a powerful database management tool that allows us to create and manage databases with ease. You can download Navicat from their official website at HTTPS://WWW.NAVICAT.COM/EN/DOWNLOAD/NAVICAT-PREMIUM. Follow the instructions provided by the installer to complete the installation process.



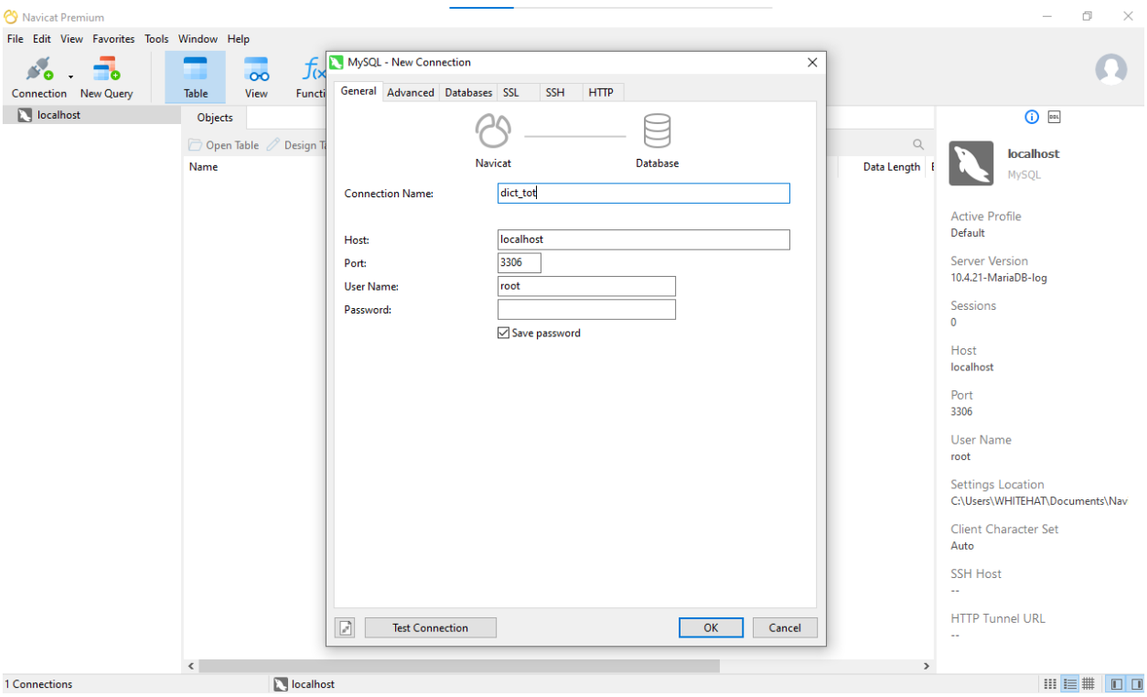
Establishing Connection

After installing Navicat, we need to establish a connection to our database server. Open Navicat and click on "New Connection" to create a new connection profile. Enter the necessary details such as the database server name, port number, username, and password. Once done, click on "Test Connection" to check if the connection is successful.

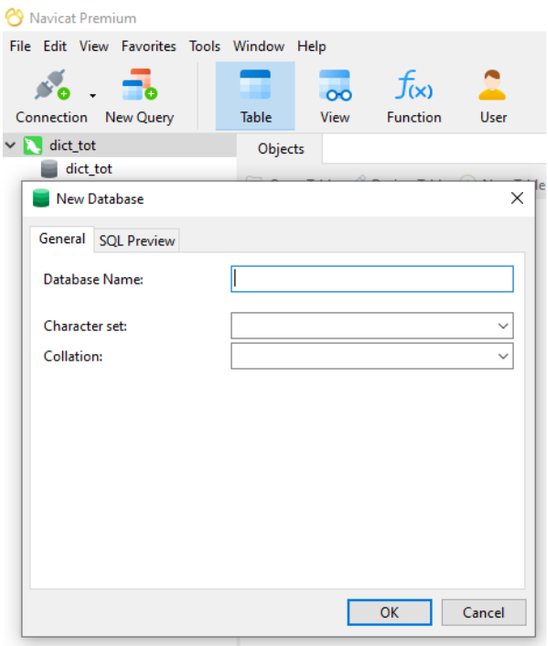
Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io



Creating Database



Once the connection is established, we can now create a new database. Right-click on the database server and select "New Database". Enter the name of the database, in this case, we will name it "dict_tot". Click on "Create" to create the database.

Creating Table

Now that we have created our database, we can now create a table. Right-click on the database and select "New Table". Enter the necessary details such as the name of the table, the number of columns, and the data types of the columns. For this LGU system, we will create a table named "demoentity" with the following columns:

Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

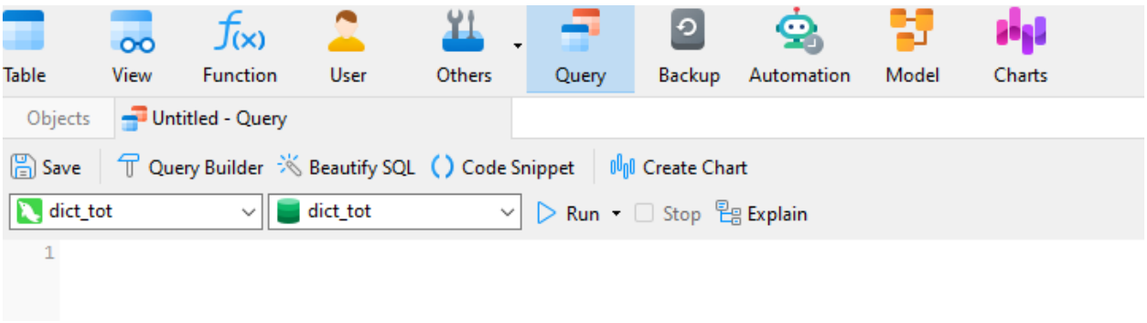
Column Name	Data Type
id	int
fname	varchar
mname	varchar
lname	varchar
suffix	varchar
sex	varchar
civilstatus	varchar
birthdate	date
height	float
weight	float
bloodtype	varchar(5)
barangay	varchar(50)
citymun	varchar(50)
province	varchar(50)
email	varchar(100)
contactno	varchar(20)
contactno2	varchar(20)
isvaccinated	tinyint
vaccinedetails	text
disease	varchar(50)
symptoms	text
medicationdetails	text
recommendation	text
dtcreated	datetime
dtupdated	datetime
remarks	text
datastatus	tinyint
temp	float
coviddiagnosed	tinyint
covidencounter	tinyint
country	varchar(50)

Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io

Once the table has been created, you can start inserting data into it using SQL queries or through the Navicat interface.



In the Navicat interface, you can easily insert data into your table by opening the table and clicking on the "Insert" button. From there, you can enter values for each field and save the record.

With this table and database set up, you can now start building your LGU system and storing user data in a secure and organized way.

Contact Us :

Name: Ronald E. Chipoco
Email: ron.chipoks1981@gmail.com
Phone: +63 912-418-7546
Github: ron-chips.github.io/ronchipocoproject.github.io

Name: Von Microbert T. Miones
Email: vonmiones@gmail.com
Phone: +63 948-850-2995
Github: vonmiones.github.io