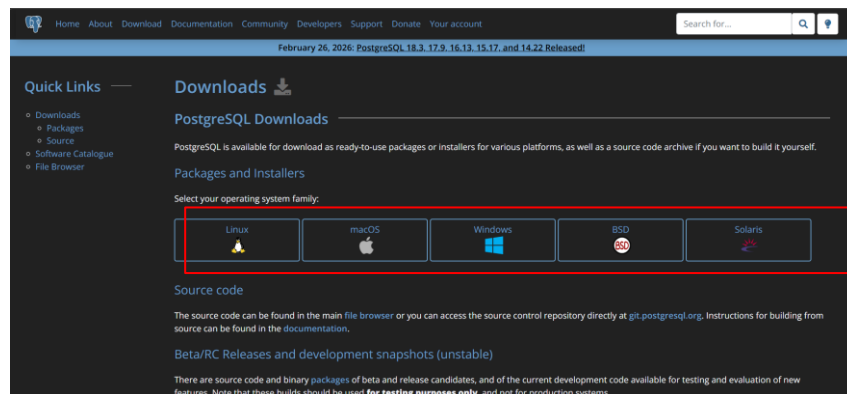


ภาคผนวก ก
คู่มือการติดตั้งโปรแกรม

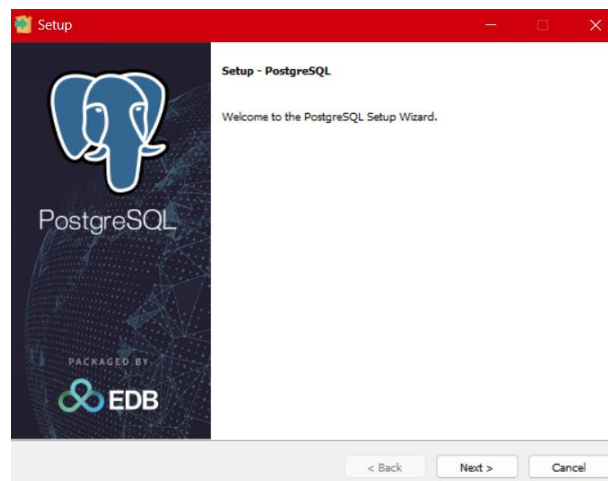
1. วิธีการติดตั้ง PostgreSQL และ pgAdmin 4

1.1 เข้าไปที่เว็บไซต์ <https://www.postgresql.org/download/> เพื่อ Download ตัวโปรแกรมสำหรับจัดการฐานข้อมูล



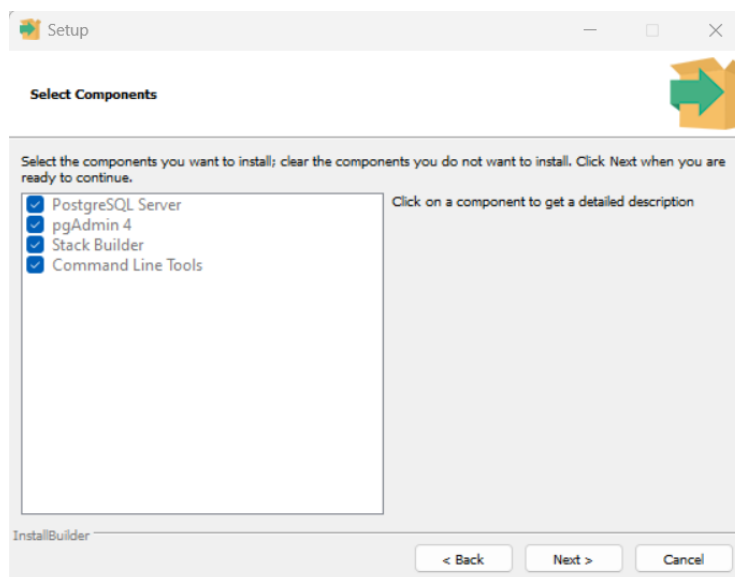
ภาพที่ ก.1 แสดงหน้าเว็บไซต์ดาวน์โหลด PostgreSQL

1.2 เมื่อทำการดาวน์โหลดเสร็จสิ้น ให้เปิดไฟล์ติดตั้ง (Installer) ขึ้นมา จะปรากฏหน้าต่างสำหรับการเริ่มการติดตั้ง ให้กดปุ่ม Next



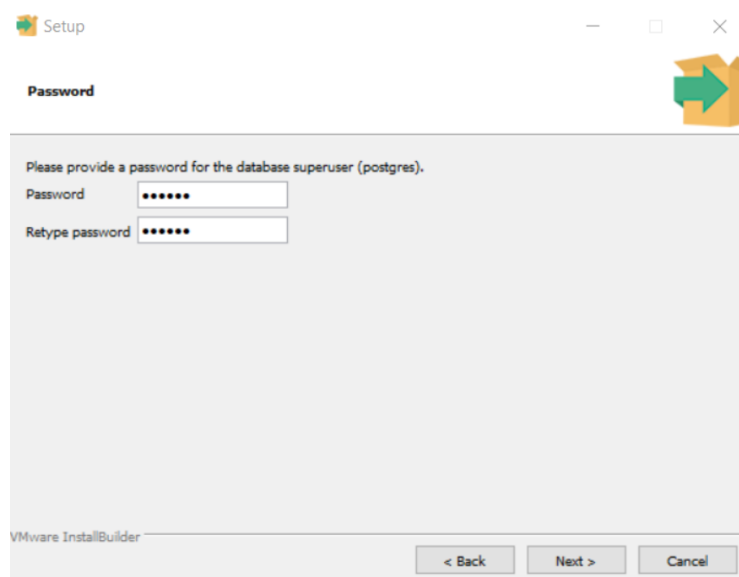
ภาพที่ ก.2 แสดงหน้าต่างเริ่มการติดตั้ง PostgreSQL

1.3 เลือกส่วนประกอบที่ต้องการติดตั้ง (ให้เลือก PostgreSQL Server, pgAdmin 4 และ Command Line Tools) จากนั้นกดปุ่ม Next



ภาพที่ ก.3 แสดงหน้าต่างเลือกส่วนประกอบที่ต้องการติดตั้ง

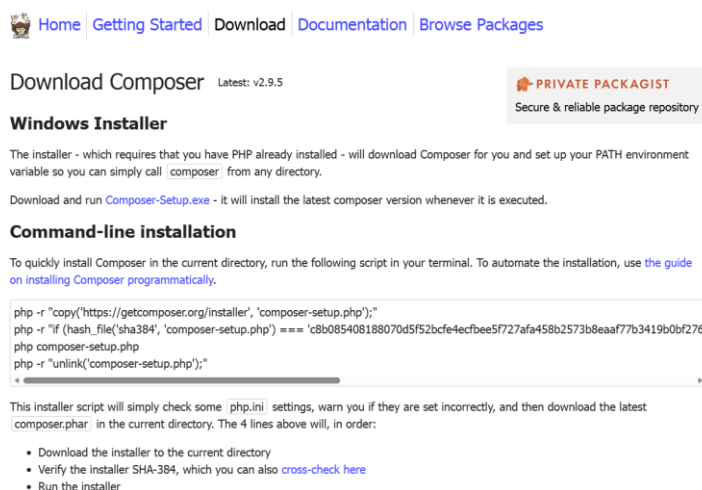
1.4 กำหนดรหัสผ่าน (Password) สำหรับ Superuser (postgres) ซึ่งรหัสผ่านนี้จะถูกนำไปใช้เชื่อมต่อฐานข้อมูลในโปรเจกต์ จากนั้นกด Next จนการติดตั้งเสร็จสมบูรณ์



ภาพที่ ก.4 แสดงหน้าต่างการตั้งค่างรหัสผ่านฐานข้อมูล

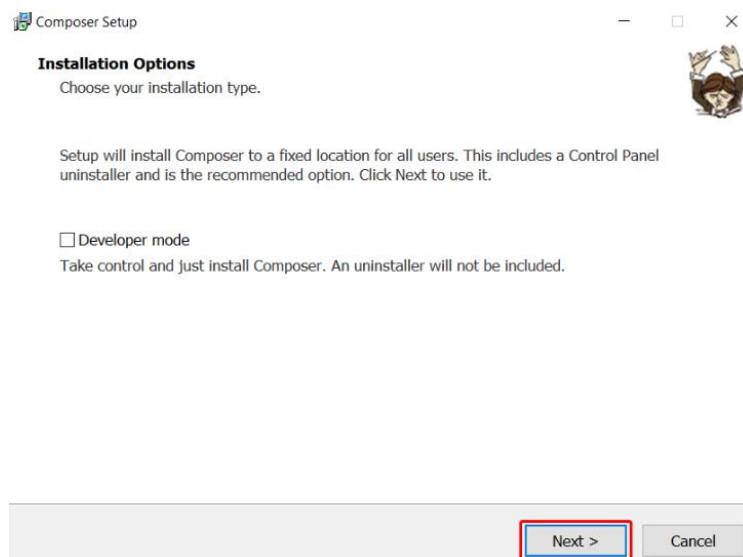
2. วิธีการติดตั้ง Composer (สำหรับใช้งาน Framework Laravel)

2.1 เข้าไปที่เว็บไซต์ <https://getcomposer.org/download/> เพื่อ Download ตัวโปรแกรมติดตั้ง Composer-Setup.exe



ภาพที่ ก.5 แสดงหน้าเว็บไซต์ดาวน์โหลด Composer

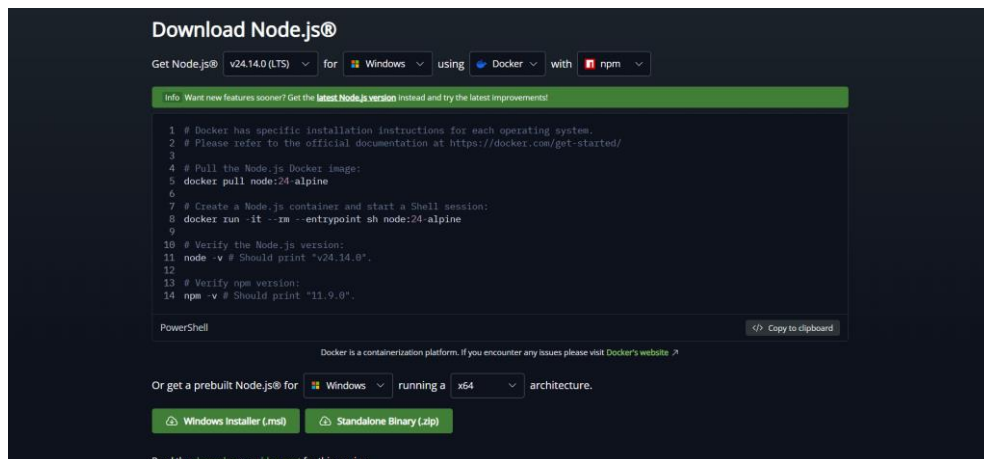
2.2 เมื่อดาวน์โหลดเสร็จแล้ว ทำการเปิดไฟล์ติดตั้ง เลือก Install for all users และกด Next ไปตามขั้นตอนมาตรฐานจนการติดตั้งเสร็จสิ้น



ภาพที่ ก.6 แสดงหน้าต่างการติดตั้ง Composer

3. วิธีการติดตั้ง Node.js (สำหรับรัน Tailwind CSS และ Alpine.js)

3.1 เข้าไปที่เว็บไซต์ <https://nodejs.org/> เพื่อทำการดาวน์โหลด Node.js เวอร์ชัน LTS (Long Term Support)



ภาพที่ ก.7 แสดงหน้าเว็บไซต์ดาวน์โหลด Node.js

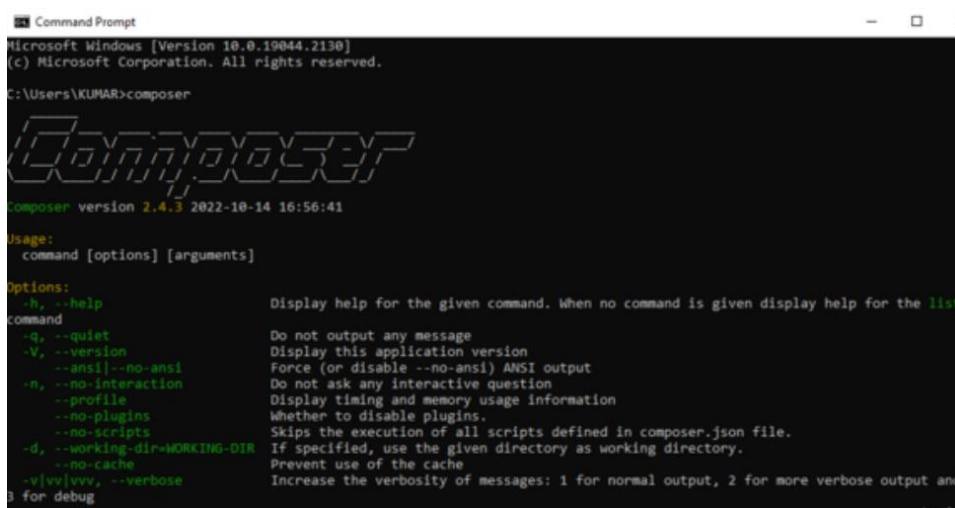
3.2 เปิดไฟล์ติดตั้งที่ดาวน์โหลดมา และกด Next ตามขั้นตอนของระบบจนกระทั่งติดตั้งโปรแกรมเสร็จสิ้น



ภาพที่ ก.8 แสดงหน้าต่างการติดตั้ง Node.js

4. วิธีการติดตั้งโปรเจกต์ Laravel และ Tailwind CSS

4.1 เปิดโปรแกรม Terminal (หรือ Command Prompt) พิมพ์คำสั่ง `composer create-project laravel/laravel watakacha` เพื่อทำการสร้างโปรเจกต์ระบบเช่าชุด



```

Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KUMAR>composer

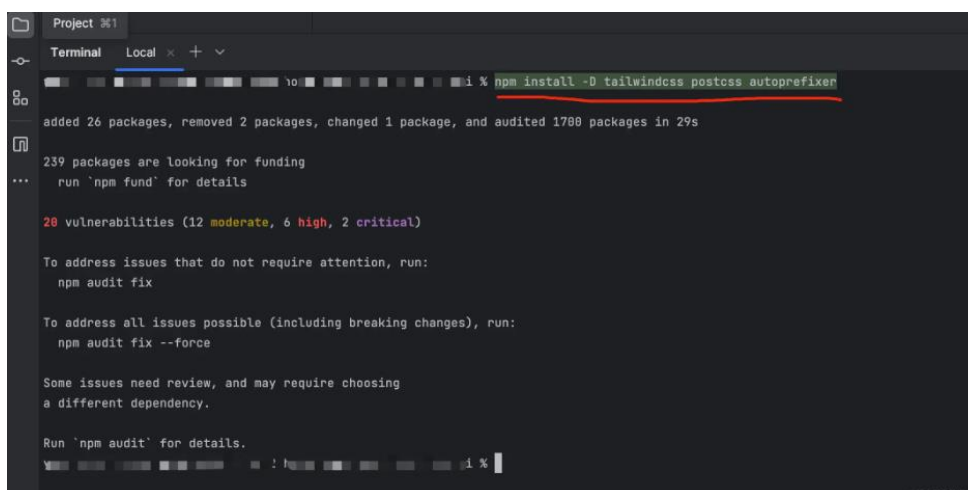
Composer version 2.4.3 2022-10-14 16:56:41

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  --no-scripts             Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
  
```

ภาพที่ ก.9 แสดงการรันคำสั่งสร้างโปรเจกต์ Laravel

4.2 เมื่อโหลดโปรเจกต์เสร็จสิ้น ให้เข้าไปที่โฟลเดอร์โปรเจกต์ด้วยคำสั่ง `cd watakacha` และทำการติดตั้ง Tailwind CSS ด้วยคำสั่ง `npm install -D tailwindcss postcss autoprefixer` ตามด้วย `npx tailwindcss init -p`



```

Project #1
Terminal Local x +
i % npm install -D tailwindcss postcss autoprefixer
added 26 packages, removed 2 packages, changed 1 package, and audited 1788 packages in 29s
239 packages are looking for funding
  run `npm fund` for details
28 vulnerabilities (12 moderate, 6 high, 2 critical)
To address issues that do not require attention, run:
  npm audit fix
To address all issues possible (including breaking changes), run:
  npm audit fix --force
Some issues need review, and may require choosing
a different dependency.
Run `npm audit` for details.
  
```

ภาพที่ ก.10 แสดงการติดตั้ง Tailwind CSS ในโปรเจกต์

4.3 พิมพ์คำสั่ง `npm run dev` เพื่อทำการคอมไพล์ไฟล์สไตล์ (CSS) และสคริปต์ต่างๆ ของระบบหน้าบ้าน

```
PS E:\watakacha\watakacha-supabase> npm run dev
> dev
> vite

VITE v7.1.12 ready in 840 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

LARAVEL v12.36.1 plugin v2.0.1

→ APP_URL: http://127.0.0.1:8000
16:51:17 [vite] full reload resources\views\member\profile.blade.php
16:51:29 [vite] full reload resources\views\member\history.blade.php
16:51:37 [vite] full reload resources\views\member\history.blade.php
```

ภาพที่ ก.11 แสดงการรันระบบคอมไพล์ไฟล์ Frontend

4.4 เปิด Terminal ขึ้นมาอีก 1 หน้าต่าง แล้วพิมพ์คำสั่ง `php artisan serve` เพื่อเปิดเซิร์ฟเวอร์จำลอง จากนั้นสามารถเข้าดูเว็บไซต์ได้ที่ `http://127.0.0.1:8000`

```
PS E:\watakacha\watakacha-supabase> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2026-02-28 16:29:14 / .....
2026-02-28 16:29:14 .....
2026-02-28 16:30:38 / .....
2026-02-28 16:30:42 .....
```

ภาพที่ ก.12 แสดงการรันเซิร์ฟเวอร์ของ Laravel

5. วิธีการตั้งค่า LINE Official Account และ LINE LIFF API

5.1 เข้าไปที่เว็บไซต์ <https://developers.line.biz/> และทำการเข้าสู่ระบบ (Log in) ด้วยบัญชี LINE ของผู้พัฒนา

[กลับไปบริการ](#)

ภาษาไทย



LINE Business ID

เข้าสู่ระบบด้วย LINE Business ID

บัญชี LINE

บัญชีธุรกิจ

สร้างบัญชี

ใช้ SSO

ภาพที่ ก.13 แสดงหน้าเว็บไซต์ LINE Developers

5.2 ทำการสร้าง Provider ใหม่ และสร้าง Channel ประเภท Messaging API เพื่อสร้าง บัญชี LINE OA สำหรับแจ้งเดือนสถานะการเข้าสู่ชุด

The image shows a three-step progress bar at the top: 1. Enter company info (active), 2. Check application, and 3. Application complete. Below this is the 'Create a LINE official account' form, which is marked as 'Required'. The form is divided into 'Login info' and 'Account info' sections.

Login info

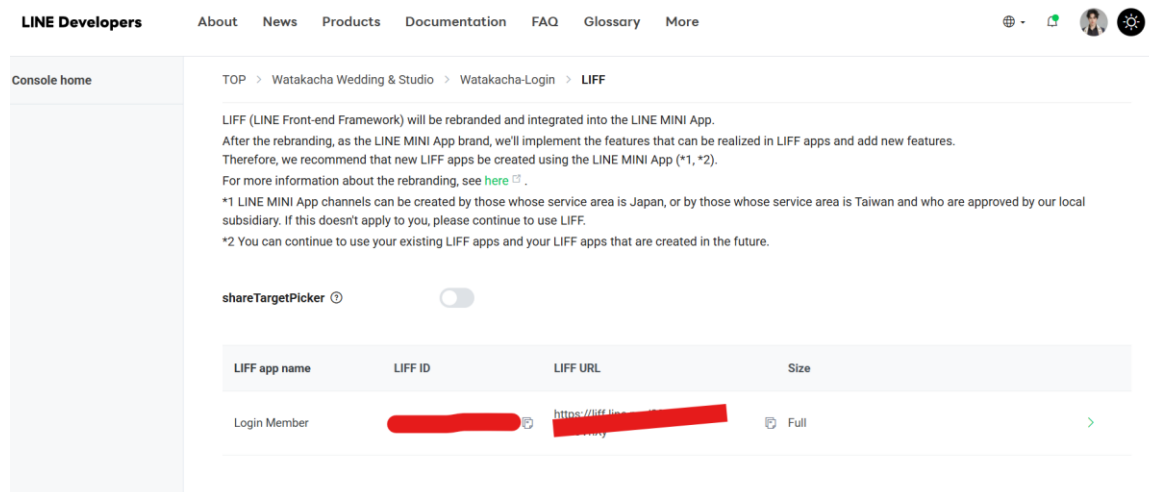
- Username: Sample [Log out](#)
- Service region: Zambia

Account info

- Account name: Ex: BROWN Cafe (0/20). This name will appear on the LINE friend list and chat screen.
- Email address: user@example.com (16/240)
- Company or owner's country or region: (Dropdown menu). This country/region info will be displayed where users can see, such as on your account profile.
- Company name: (Text input field, 0/100)
- Industry: (Main industry dropdown) and (Subindustry dropdown)

ภาพที่ ก.14 แสดงหน้าต่างการสร้าง Channel Messaging API

5.3 ทำการสร้าง Channel ประเภท LINE LOGIN เพื่อตั้งค่าใช้งานระบบ LIFF (LINE Front-end Framework) และนำ URL ของเว็บไซต์ไปตั้งค่าในช่อง Endpoint URL



ภาพที่ ก.15 แสดงการตั้งค่า Endpoint URL สำหรับระบบ LIFF

5.4 คัดลอกค่า Channel Access Token และ Channel Secret มาใส่ในไฟล์ .env ของโปรเจกต์ Laravel เพื่อให้ระบบหลังบ้านสามารถเชื่อมต่อและส่งข้อความแจ้งเตือนไปยังแชทของลูกค้านได้

```

.env
register.blade.php
LineWebhookController.php

.env
61
62 AWS_ACCESS_KEY_ID=
63 AWS_SECRET_ACCESS_KEY=
64 AWS_DEFAULT_REGION=us-east-1
65 AWS_BUCKET=
66 AWS_USE_PATH_STYLE_ENDPOINT=false
67
68 VITE_APP_NAME="{APP_NAME}"
69
70 LINE_CHANNEL_ACCESS_TOKEN="[Redacted]"
71 LINE_CHANNEL_SECRET="[Redacted]"
72 LIFF_ID="[Redacted]"
73 ## แชนแนล ID (Channel ID) watakacha
74 LINE_CHANNEL_ID="[Redacted]"
75 ## ความลับแชนแนล (Channel Secret)
76 # LINE_CHANNEL_SECRET="[Redacted]"

```

ภาพที่ ก.16 แสดงการนำค่า API Key มาตั้งค่าในระบบ