FPT ACADEMY INTERNATIONAL

FPT – APTECH COMPUTER EDUCATION

COURIERXPRESS PROJECT SETUP AND DEPLOYMENT GUIDE

Supervisor: Le Mong Thuy

Batch: T1.2406.E1

Group: 2

Student members:

Nguyen Hoang Hiep - Student1685539

Dao Nguyen Phuoc - Student1685498

Pham Kim Nghi - Student1685515

Le Hong Duc Plinl - Student1686149

Ho Chi Minh city, Vietnam

01/2026

Abstract

This document provides a structured and reproducible procedure for installing and running the CourierXpress project on a new workstation. The guide covers environment prerequisites, backend (Laravel) configuration, database provisioning, seeding for end-to-end demonstrations, frontend (React/TypeScript) setup, default test accounts, and common troubleshooting practices.

Table of Contents

1. Scope and Assumptions

- This guide targets local development and demonstration environments.
- Commands are presented for Windows, Linux, and macOS where differences are relevant.
- The project repository contains two primary directories: backend/ (Laravel) and frontend/ (React).
- The user has basic familiarity with terminal/command prompt usage.

2. System Requirements

Table 1 summarizes the minimum software requirements to build and run the CourierXpress backend and frontend.

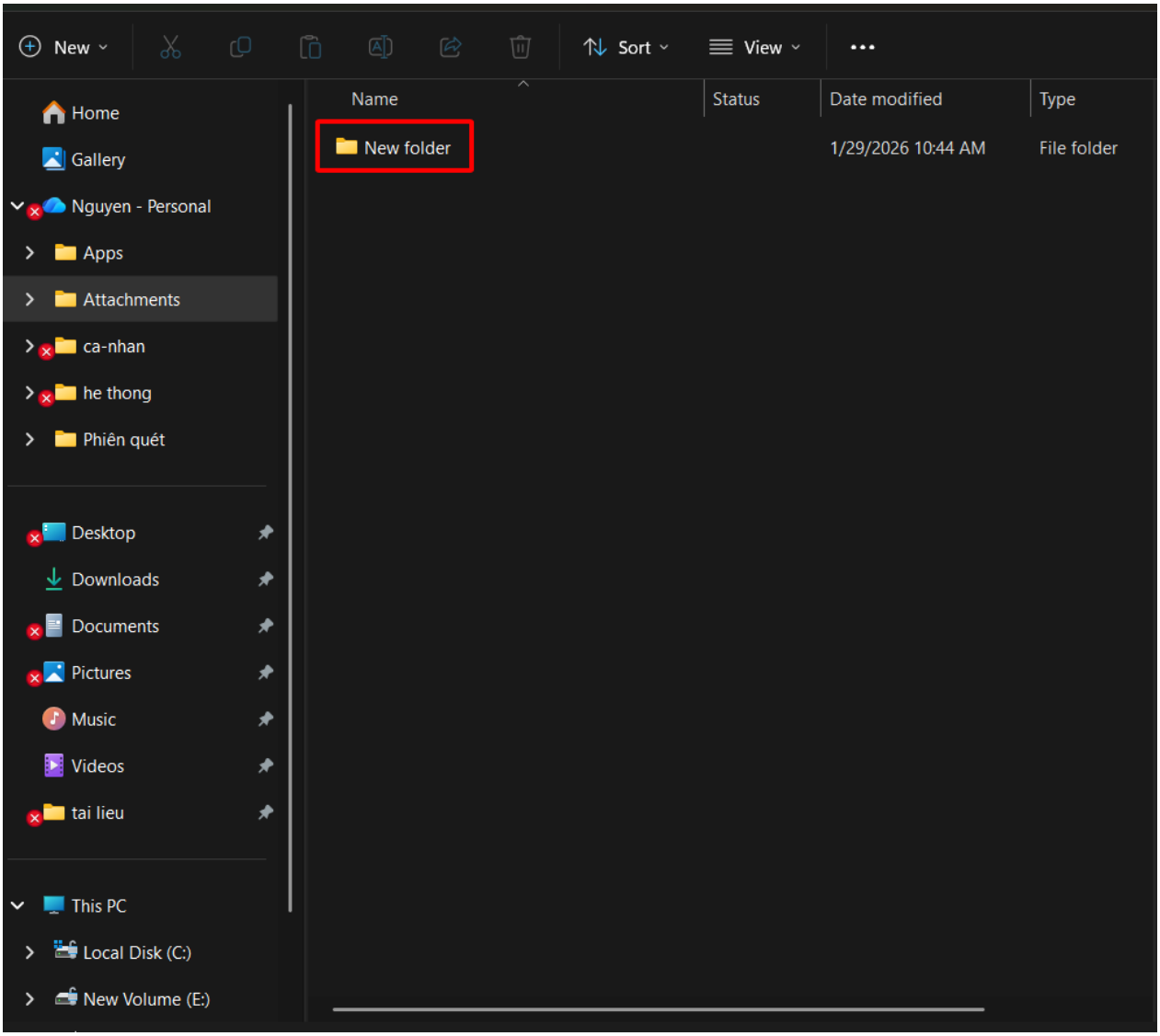
Component	Requirement	Purpose/Notes
Backend (Laravel)	PHP 8.2 or later	Required to run Laravel application code.
Backend (Laravel)	Composer	Dependency manager for PHP packages.
Database	MySQL 5.7+ or MariaDB 10.3+	Relational database for persistent storage.
Frontend Tooling	Node.js 18.x or later	Required to run Vite/React toolchain and build assets.
Frontend Tooling	npm 9.x+ or Yarn 1.22+	JavaScript package manager.
Administration Tool (Optional)	phpMyAdmin	Convenient database management (optional).

Table 1. Software requirements

4. Project Clone and Directory Validation

1. Choose a location to store the project. For example: D:\Projects\ (Windows) or ~/Projects/ (macOS/Linux).

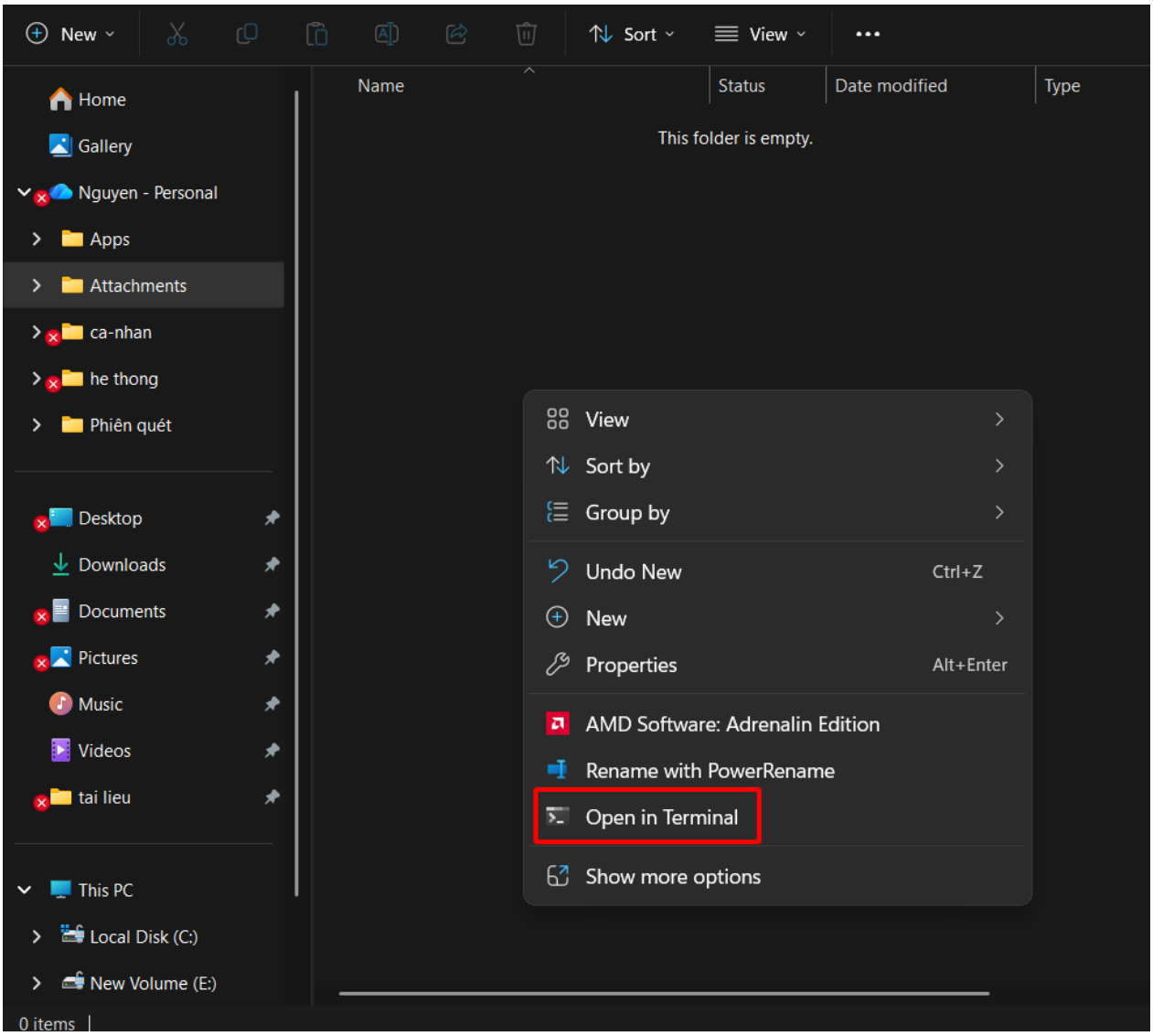
2. Open that folder in File Explorer (Windows) / Finder (macOS)



3. Right-click inside the folder (in an empty area) and select:

- **Windows 11:** *Open in Terminal*

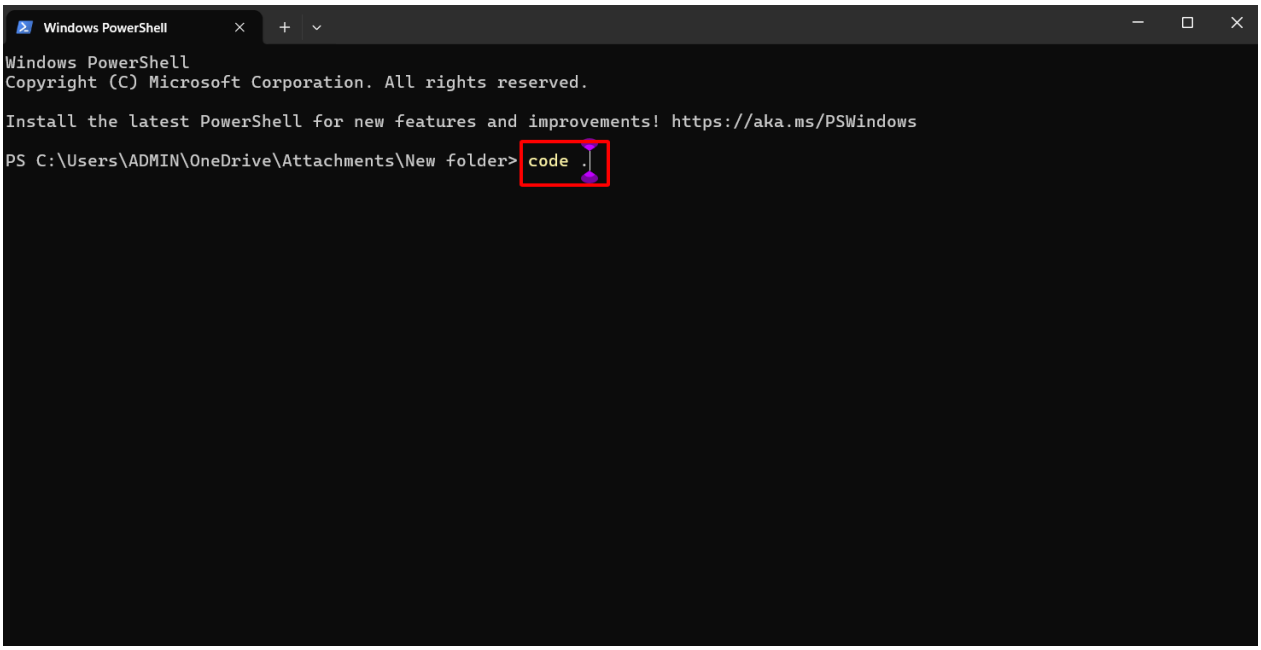
- **macOS/Linux:** Open Terminal, then cd to the folder path (or use “New Terminal at Folder” if available)



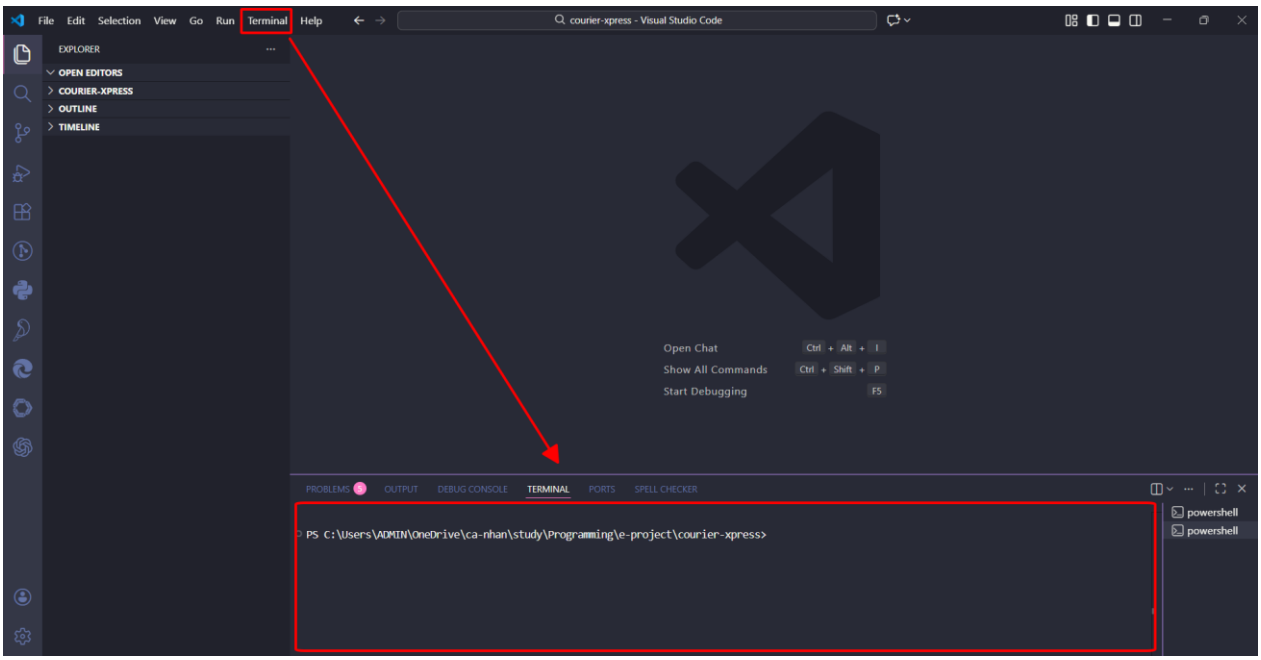
4. In the terminal, run

```
Code .
```

This command opens **the current folder** in Visual Studio Code (VS Code).



5. Inside VS Code: Go to **Terminal → New Terminal**. This opens a terminal panel at the bottom of VS Code, typically already pointing to the current project directory.



6. In the VS Code terminal, run the clone command: [hiepstudy1604-ux/courierXpress]

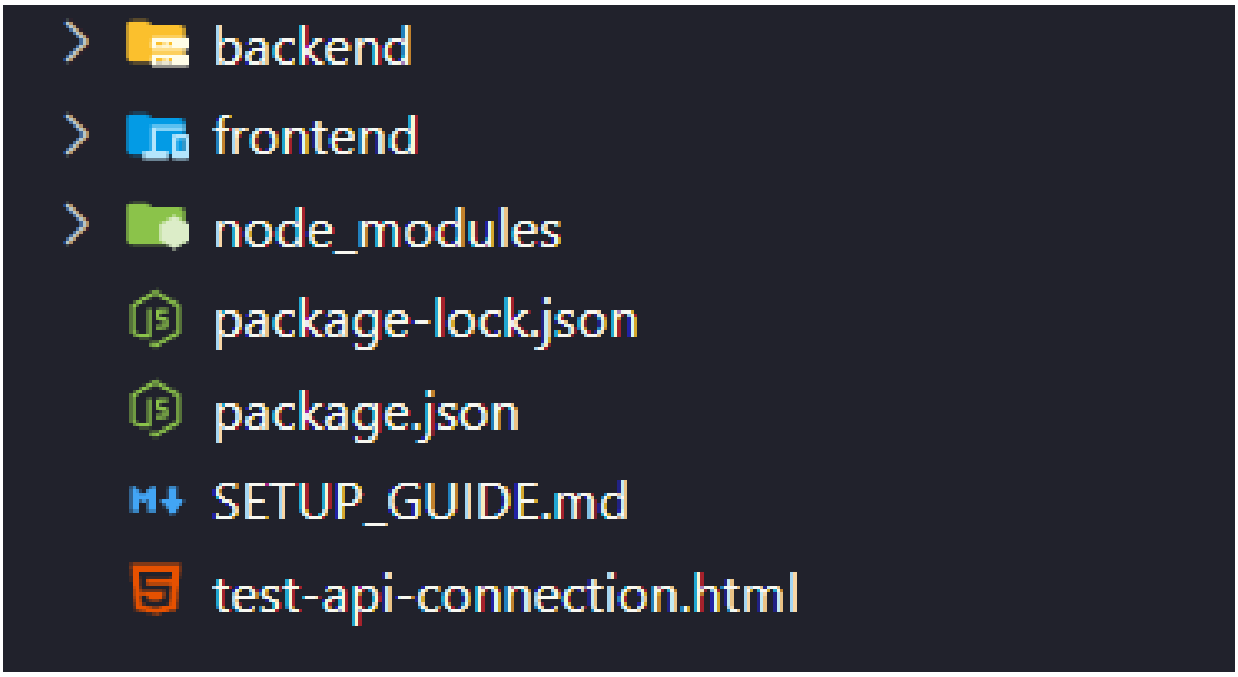
```
git clone https://github.com/hiepstudy1604-ux/courierXpress.git
```

7. Move into the project folder:

```
cd courierXpress
```

At the project folder, verify that the resulting directory structure contains the backend and frontend modules as shown below.

```
courier-xpress/
├── backend/           # Laravel backend
├── frontend/          # React/TypeScript frontend
├── SETUP_GUIDE.md     # Original setup instructions (source)
└── ...
```



4. Backend Setup (Laravel)

This section describes how to install PHP dependencies, configure environment variables, provision the database, migrate schemas, seed demonstration data, and run the Laravel server.

4.1 Install Dependencies

From the project root folder (courierXpress), go to the Laravel backend directory

```
cd backend
```

Install PHP dependencies. This will downloads and installs all backend libraries required by the Laravel application (vendor packages).

```
composer install
```

4.2 Configure Environment Variables

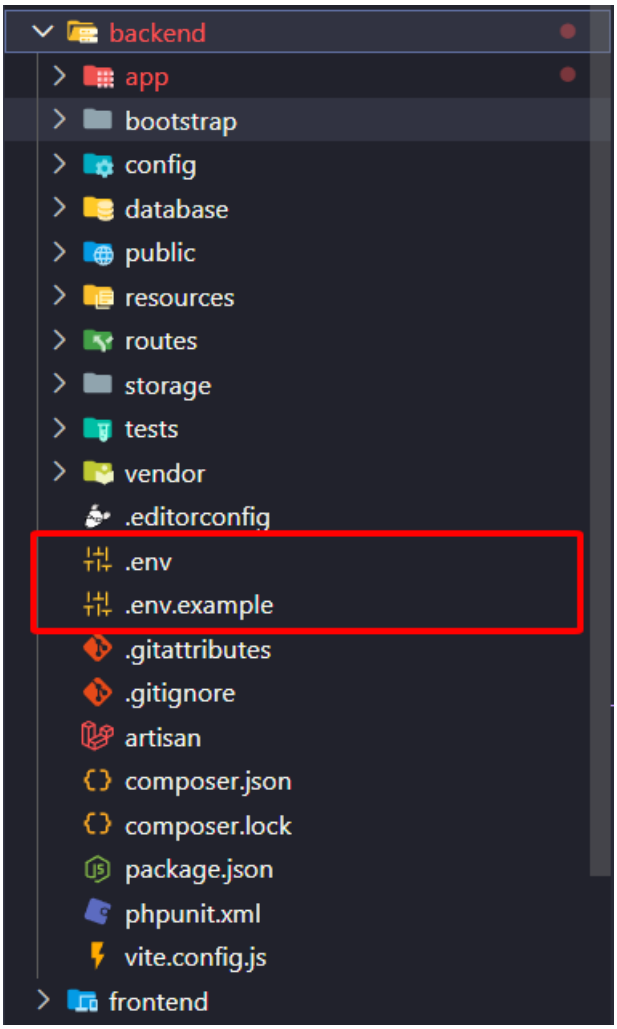
Create a local environment file by copying the template (.env.example) and generating the application key and JWT secret.

Copy the environment template:

```
# Windows (PowerShell)
Copy-Item .env.example .env

# Windows (CMD)
copy .env.example .env

# Linux/macOS
cp .env.example .env
```



In backend folder, run below command to generate the Laravel application key:

```
php artisan key:generate
```

Generate the JWT secret:

```
php artisan jwt:secret
```

4.3 Database Provisioning

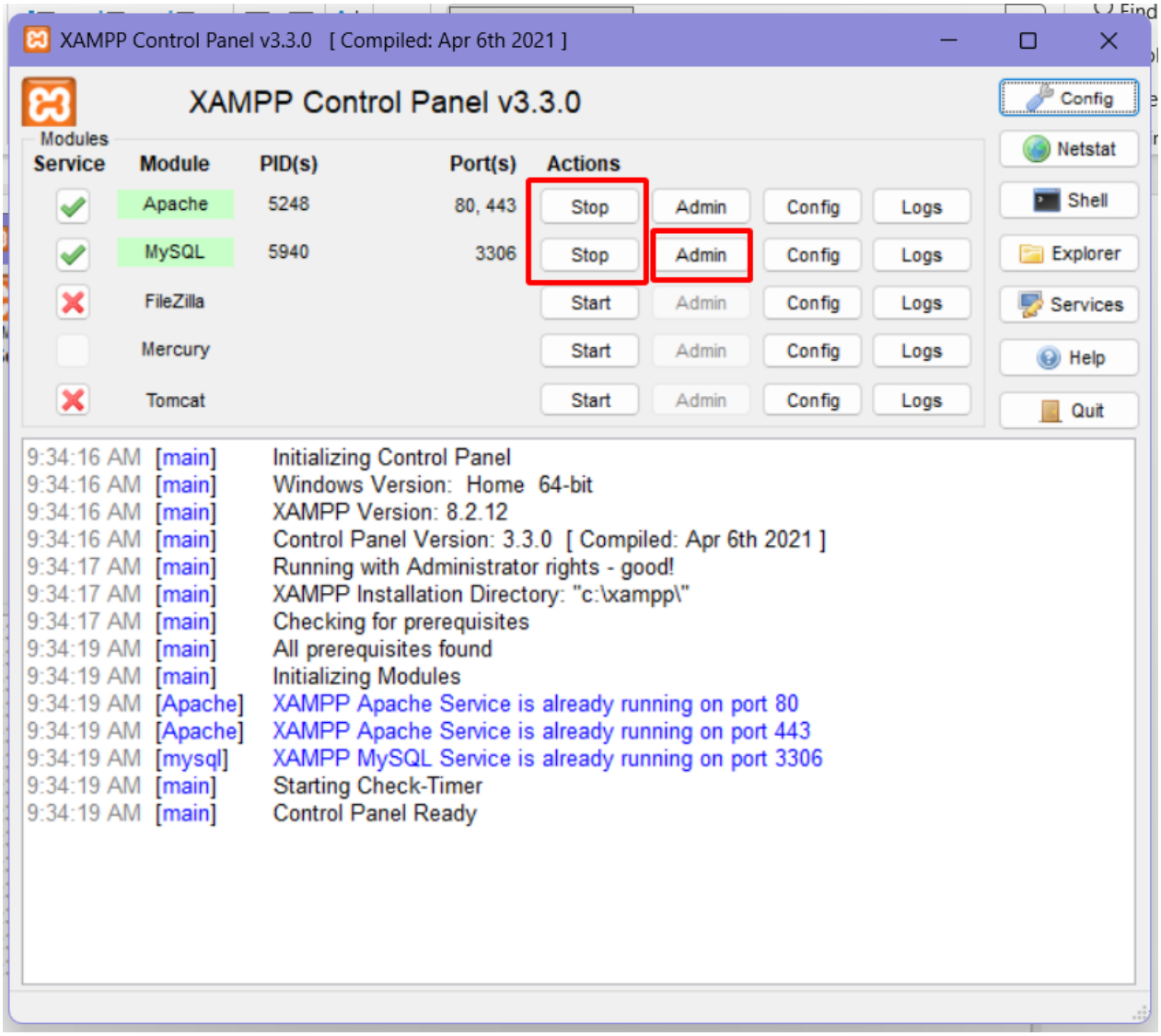
1. Open the .env file (located in the Laravel backend directory) and update the database configuration as follows

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=courier_xpress
DB_USERNAME=root
DB_PASSWORD=
DB_COLLATION=utf8mb4_unicode_ci
```

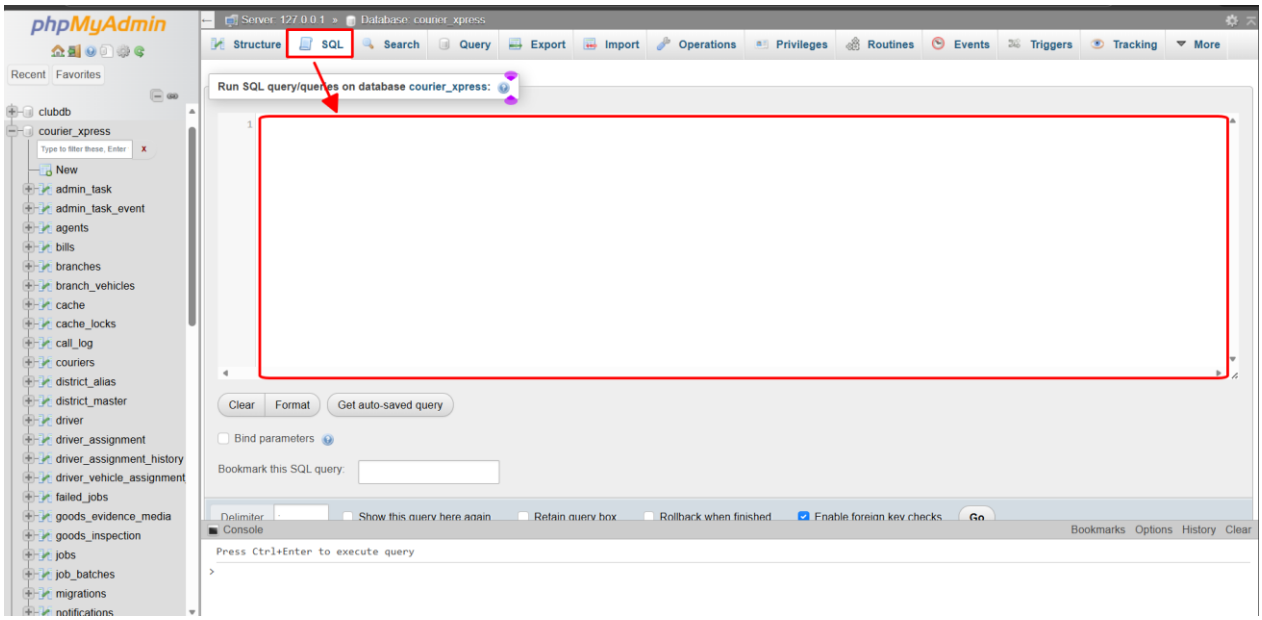
Note: Let the password local environments root user empty.

2) Create the database using phpMyAdmin (XAMPP)

1. Open **XAMPP Control Panel**.
2. Locate **MySQL**, then click **Admin** to open **phpMyAdmin** in your browser.



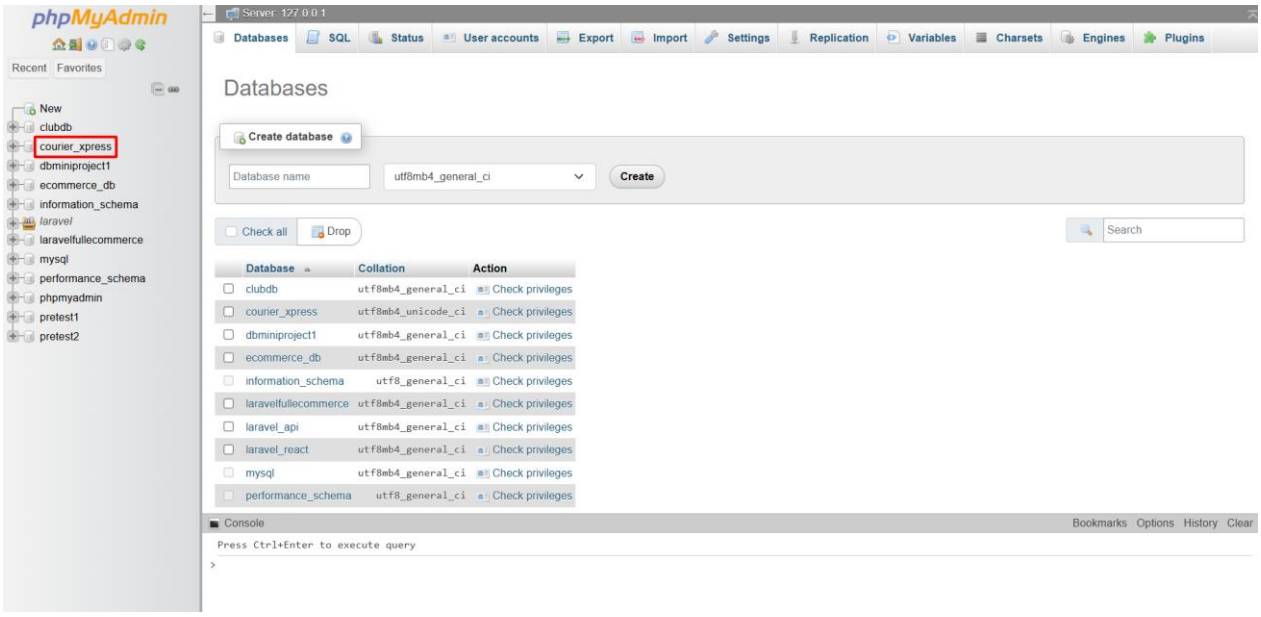
3. **In phpMyAdmin, select the SQL tab.**



4. Execute the following SQL statement:

```
CREATE DATABASE courier_xpress
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;
```

After execution, the database named **courier_xpress** will be created and available for Laravel to connect to.



4.4 Database Migration and Seeding

Migrations create the database tables required by the application. Seeding populates the database with realistic demonstration data to support end-to-end testing and dashboard analytics.

Run migrations:

In the backend directory, run:

```
php artisan migrate
```

Run the comprehensive seeder:

```
php artisan db:seed --class=ComprehensiveDatabaseSeeder
```

The comprehensive seeder is designed to generate a sizable dataset (e.g., shipments across multiple statuses over approximately 90 days, payment intents, warehouse scans, transit manifests, tasks, notifications, bills, and default user accounts) for realistic testing.

4.5 Storage Link

In the backend directory create the public storage symbolic link to enable access to uploaded files via the web server.

```
php artisan storage:link
```

4.6 Run the Backend Server

Start the Laravel development server:

```
php artisan serve
```

By default, the backend will be available at: http://localhost:8000

5. Frontend Setup (React/TypeScript)

This section describes how to install JavaScript dependencies, verify the API endpoint configuration, and run the Vite development server.

5.1 Install Dependencies

Navigate to the frontend directory and install dependencies:

```
cd frontend
npm install
npm install html2canvas jspdf
```

5.2 Configure API Endpoint

Verify that the frontend API base URL points to the running backend instance. The configuration is typically located in frontend/src/services/api.ts (or a similar configuration file).

```
// Example configuration:
const API_BASE_URL = 'http://localhost:8000/api';
```

If the backend is served on a different port or host, update API_BASE_URL accordingly.

5.3 Run the Frontend Server

Start the frontend development server:

```
npm run dev
```

By default, the frontend will be available at: http://localhost:5173

6. Service Endpoints

Service	Default URL
Frontend (Vite)	http://localhost:5173
Backend (Laravel)	http://localhost:8000
Backend API Base	http://localhost:8000/api

Table 2. Service Endpoints table

7. Default Demonstration Accounts

After seeding, the following accounts are available for demonstration and testing purposes. For security reasons, these credentials should not be used in a production environment.

Role	Email	Password	Access Scope / Notes
Admin	admin@courierxpress.com	admin123456	Full access across the system.
Agent	agent@courierxpress.com	agent123456	Branch management and shipments within

			the assigned branch.
Customer	customer@example.com	customer123	Create and track personal shipments.

Table 3. Default Demonstration Accounts

Implementation note: the agent user is typically assigned a branch during seeding (e.g., via Branch::first()).

8. Verification Checklist

Upon completion, confirm the following:

- Laravel server is running and the welcome page loads at http://localhost:8000.
- API requests from the frontend succeed without CORS errors.
- Frontend loads at http://localhost:5173 and authentication works using seeded accounts.
- Database tables are created and seeded data (shipments, notifications, tasks) is present.