**Practice Making API Using Laravel 11 and Ngrok**

*Andreika Luna Alghivari*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: andreikaluna@student.ub.ac.id*

**Abstract**

This project explores the practice of creating and testing RESTful APIs using Laravel 11 and Ngrok on the Wokwi platform within Visual Studio Code. Laravel, a robust PHP framework, simplifies the development of APIs with its built-in features, while Ngrok enables secure tunneling for local APIs to be accessed globally. The simulation demonstrates the step-by-step process of building scalable APIs, including CRUD operations, authentication, and endpoint testing. This practice provides a comprehensive understanding of API development and deployment, paving the way for real-world applications in web and mobile systems.

*Keywords: Laravel, API, Ngrok, Postman, Visual Studio Code.*

**1. Introduction**

* 1. **Background**

Laravel is one of the most popular PHP frameworks due to its elegant syntax, built-in tools for API development, and support for modern web applications. RESTful APIs facilitate communication between systems using standard HTTP methods like GET, POST, PUT, and DELETE. Ngrok complements this by creating secure tunnels that expose local APIs to the internet, enabling remote access for testing or integration purposes. Visual Studio Code with PlatformIO serves as an efficient development environment for managing code and debugging. This combination of tools offers developers a streamlined workflow for building robust APIs while ensuring accessibility during testing.

* 1. **Objective**

1. To demonstrate the creation of a RESTful API using Laravel 11
2. To integrate Ngrok for exposing local APIs to the internet securely.
3. To test API endpoints using tools like Postman or cURL for validating functionality.

**2. Methodology**

**2.1 Tools & Materials**

Laravel 11, Postman, Xampp, Ngrok, Composer, and Visual Studio Code.

**2.2 Implementation Steps**

1. Download the packages needed for laravel 11 with the command in the terminal

composer create-project --prefer-dist laravel/laravel:11.0 laravel-11

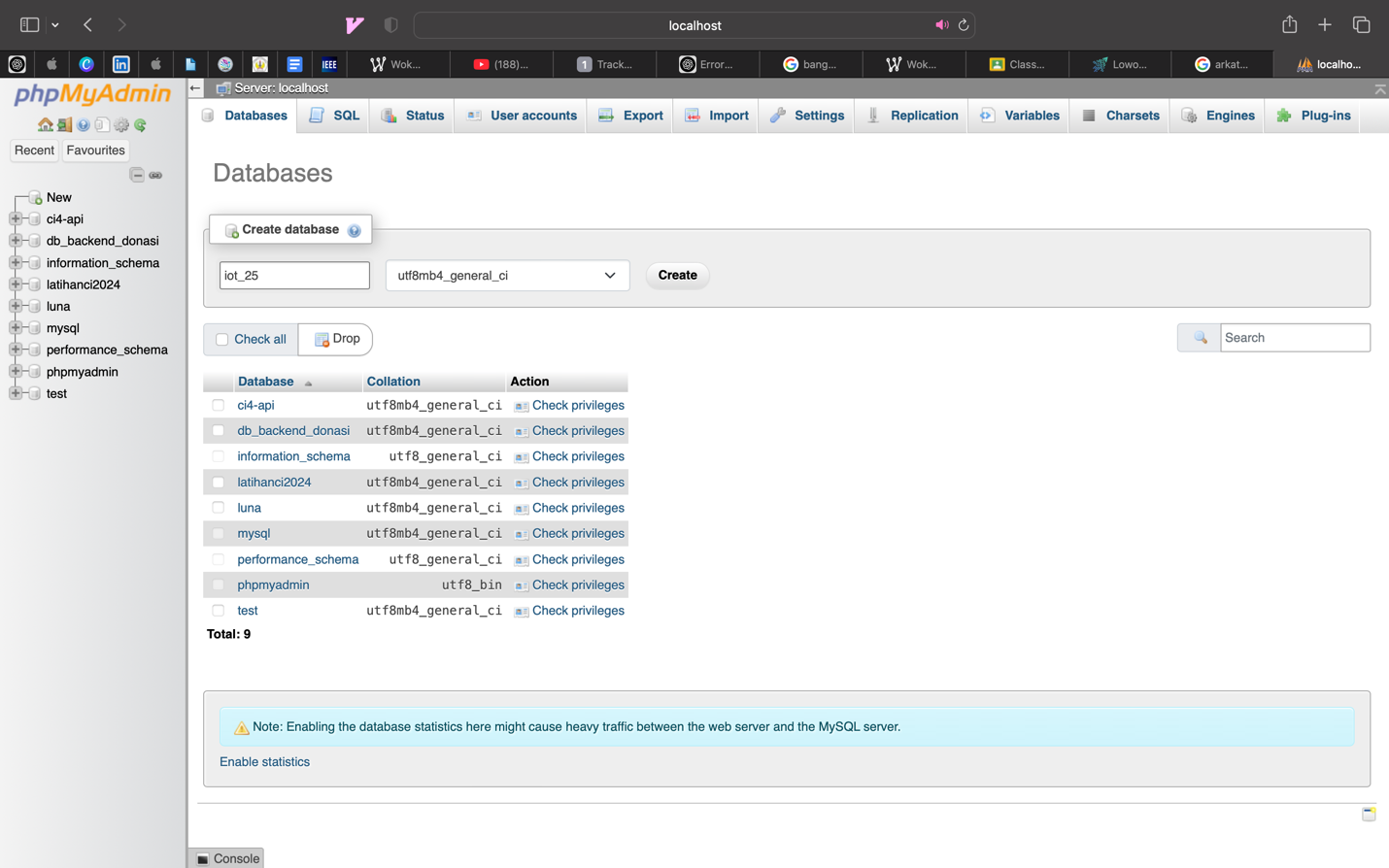
cd laravel-11

1. Create a database on phpmyadmin with the name iot\_25
2. Change the contents of the .env file configuration in the db\_username and db\_password sections according to the laptop settings
3. Create the TransactionSensor.php model file (php artisan make:model TransactionSensor -m) then change the create transaction sensors table file in the databases-migrations folder
4. Change the contents of the app/Models/TransactionSensor.php file
5. Create the table by running the php artisan migrate command on the terminal
6. Create a resource by running the command php artisan make:resource TransactionSensorResource
7. Change the contents of the TransactionSensorResource.php file in the app-Http-Resources folder
8. Create an API controller with the command php artisan make:controller Api/TransactionSensorController, then change the contents of the file
9. Create a special API route with the command php artisan install:api, then change the contents of the routes/api.php file
10. Ensure that the routes have been formed with the php artisan route:list command
11. Testing with the postman application, in the url section enter the laravel servel address http://localhost:8000/api/posts
12. Select the GET method to retrieve data from the database, then click the SEND button
13. Then try to insert data into a table in the database using the API
14. Check manually in phpmyadmin and make sure the new data enters
15. Connect the API using the ngrok service so that the API can be accessed via an iot device or wokwi iot simulation by running the ngrok command http http://localhost:8000
16. Experiment using postman using the URL given by ngrok. For GET API experiments, the URL must add the endpoint address /api/posts

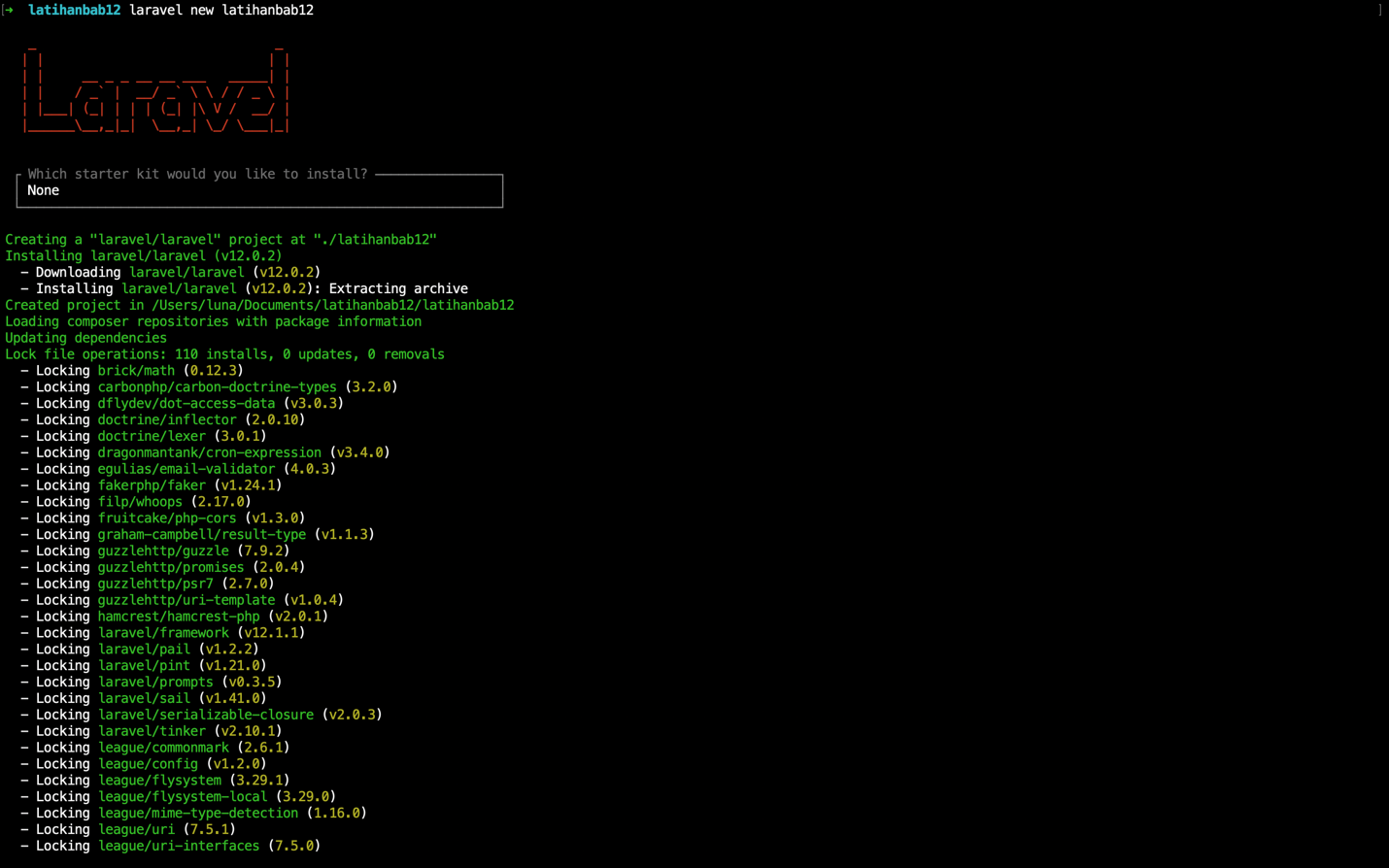
**3. Results and Discussion**

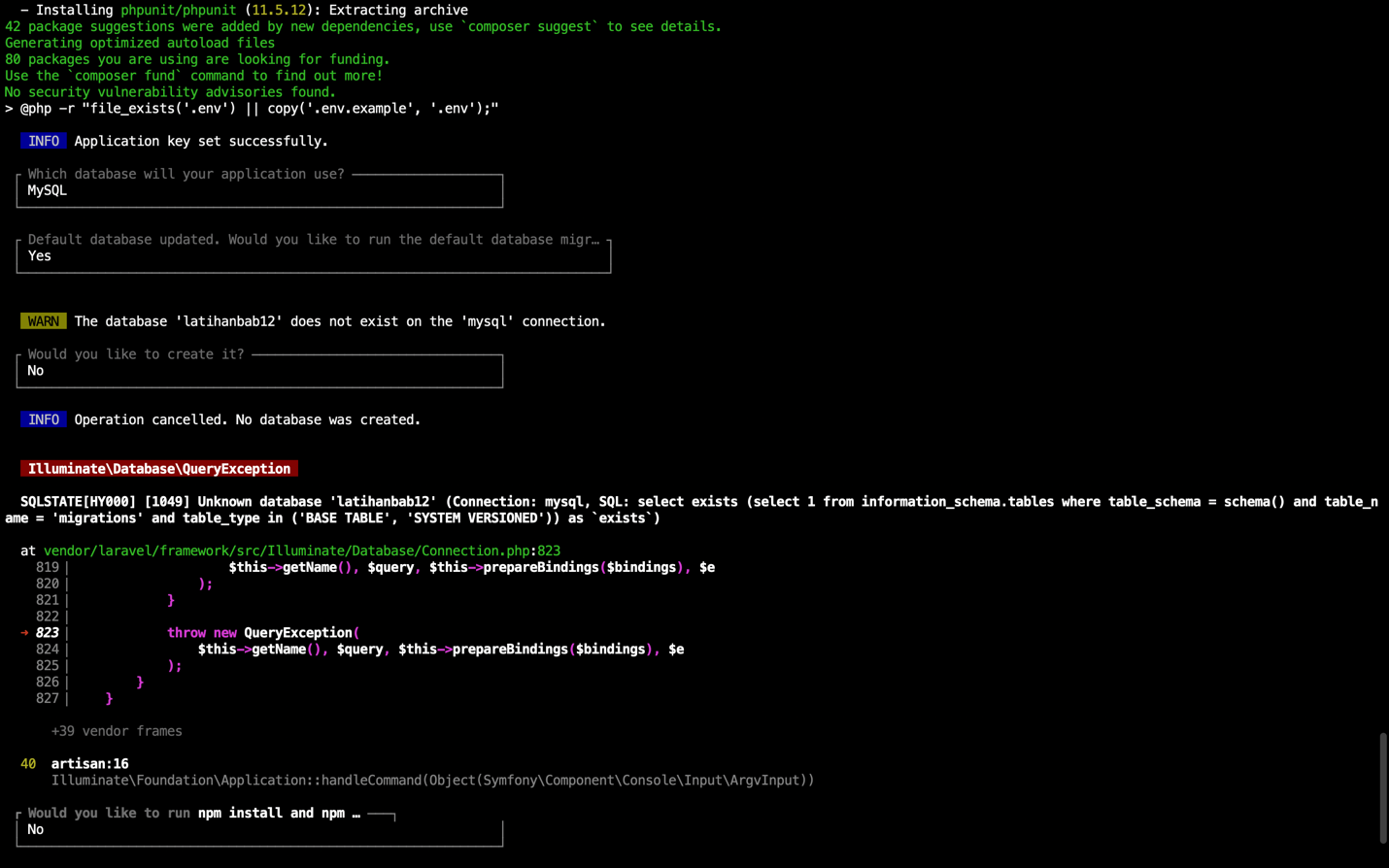
**3.1 Experimental Results**

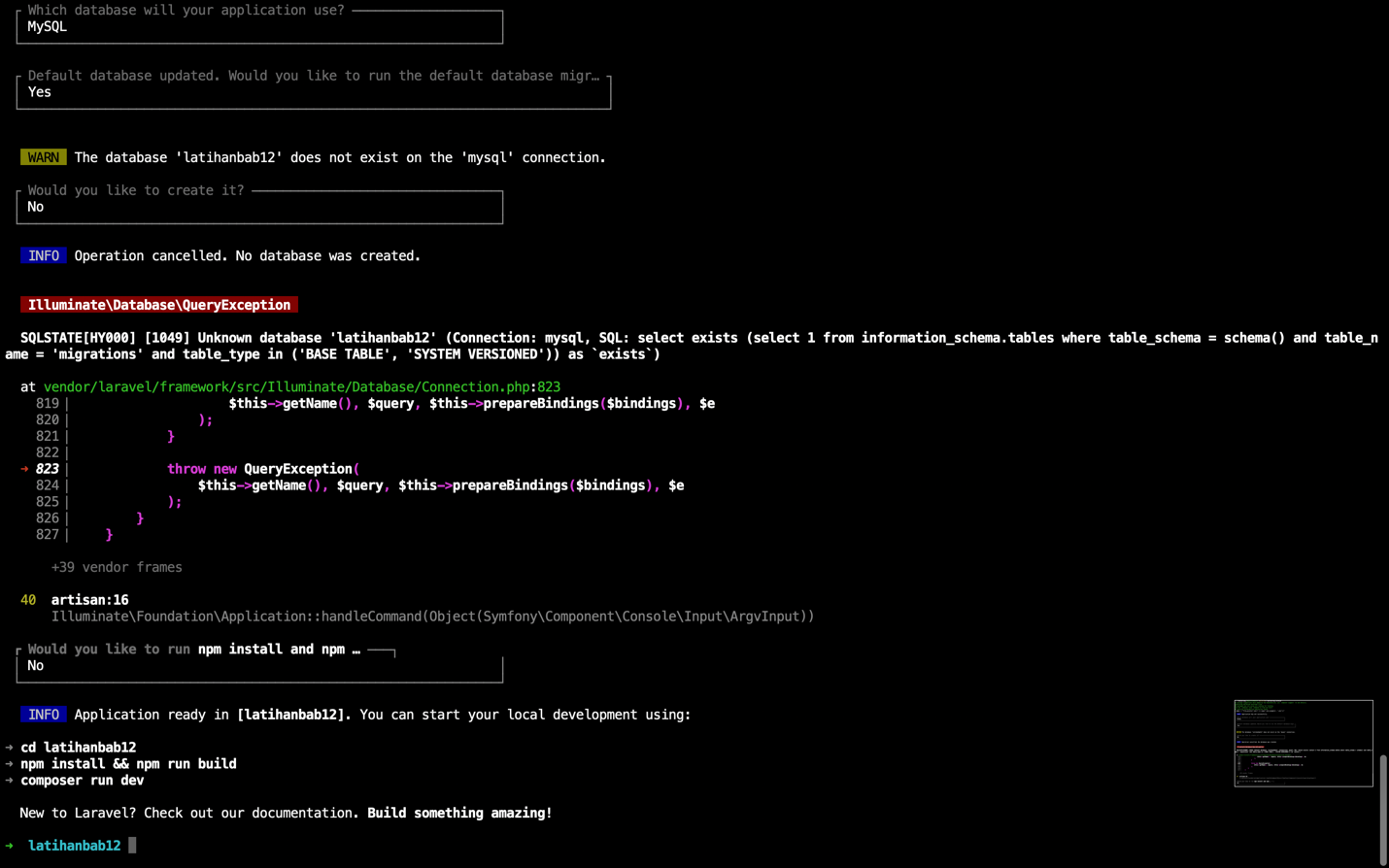
1. Make database on phpMyAdmin



1. Make folder “latihanbab12”





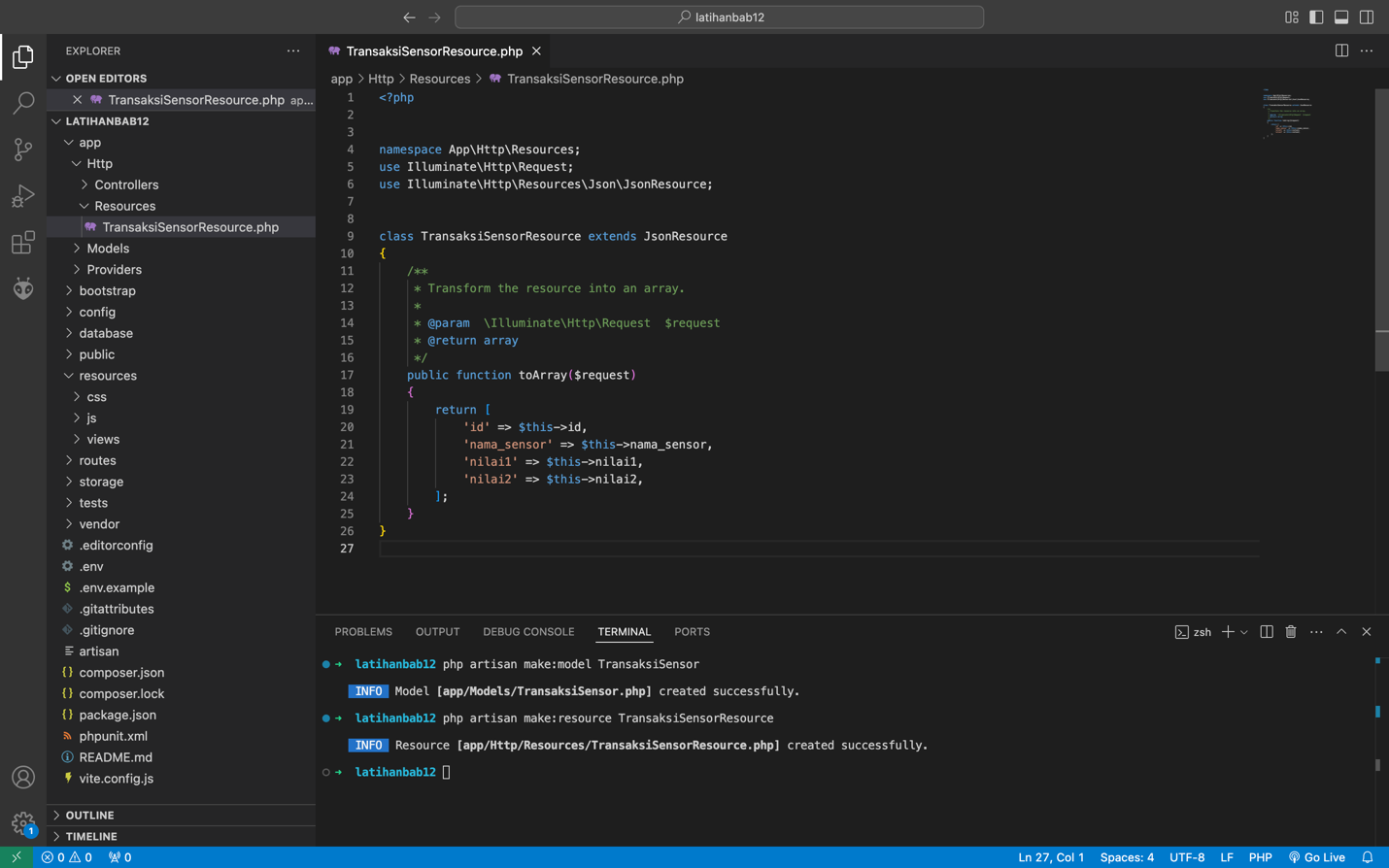


1. Create the TransactionSensor.php model file (php artisan make:model TransactionSensor)

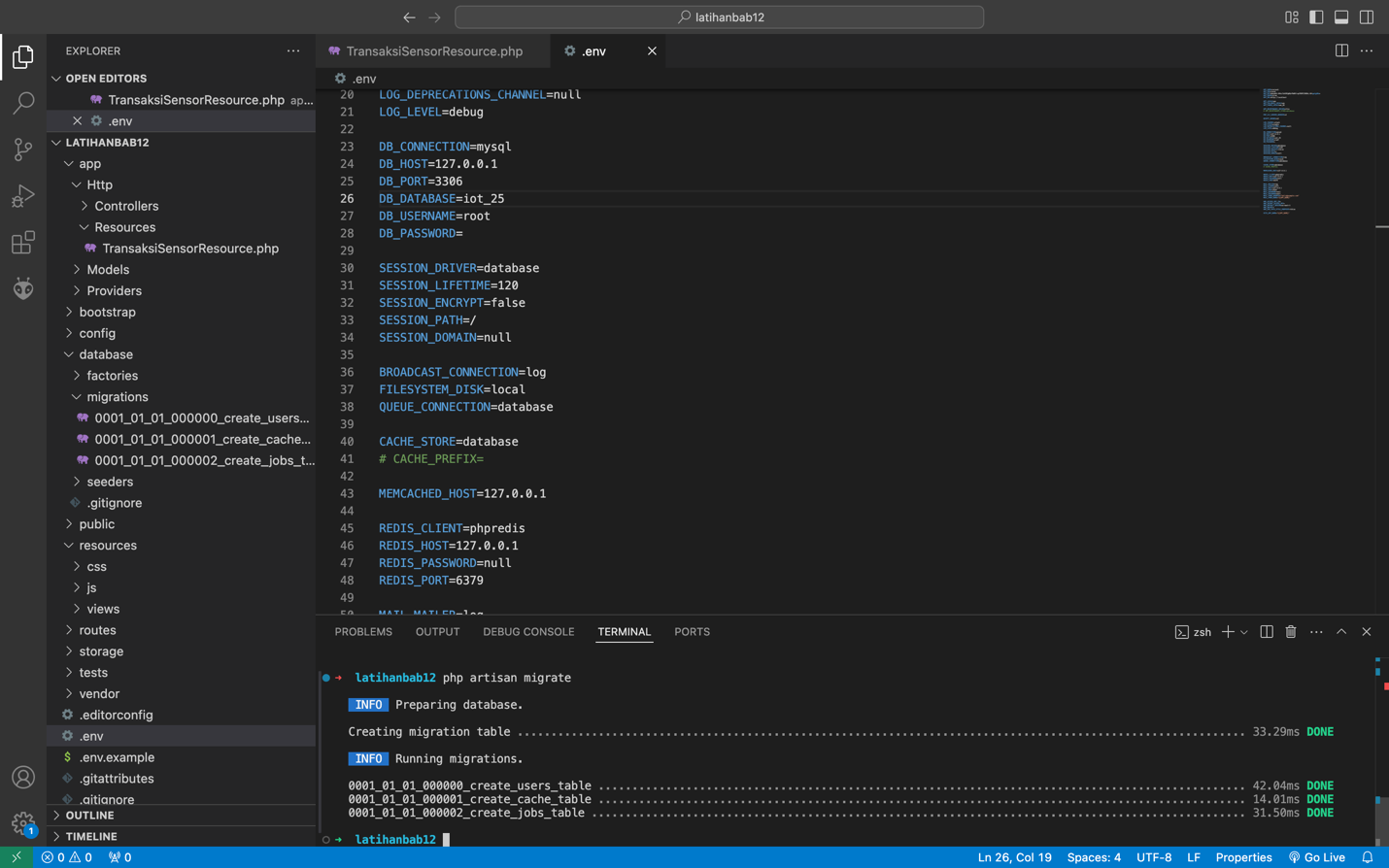
A screenshot of a computer

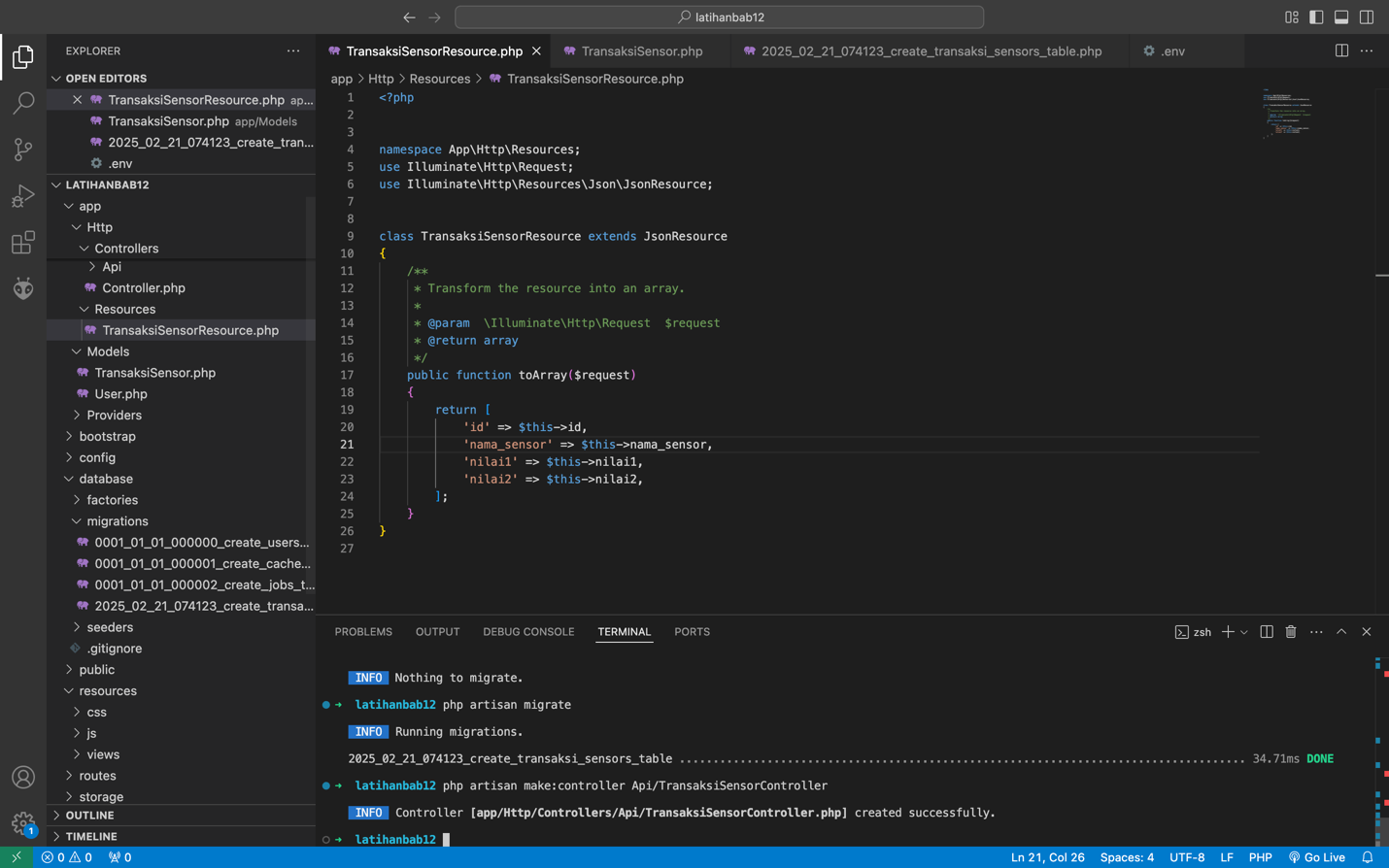
Description automatically generated

1. Create a resource by running the command php artisan make:resource TransactionSensorResource



1. Then Migrate





1. Testing with the postman application

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Check manually in phpmyadmin and make sure the new data enters

A screenshot of a computer

Description automatically generated

1. Connect the API using the ngrok service so that the API can be accessed via an iot device or wokwi iot simulation by running the ngrok command http <http://localhost:8000>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Experiment using postman using the URL given by ngrok. For GET API experiments, the URL must add the endpoint address /api/posts

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

At this stage, the API developed using Laravel 11 is functioning effectively and can be accessed through a public URL. The successful implementation of CRUD operations and authentication mechanisms demonstrates the robustness of the API. Utilizing Ngrok has enabled secure tunneling, allowing for remote access to the local development environment. This accomplishment not only validates the development process but also highlights the potential for deploying the API in real-world applications. Moving forward, this API can serve as a foundational component for various web and mobile applications, facilitating seamless integration and communication between systems.