Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образование учреждение высшего образования

«Северо-Восточной федеральной университет им. М. К. Аммосова»

Колледж инфраструктурных технологий

Кафедра эксплуатации и обслуживания информационных систем

**Тестовое Задание для Backend Разработчика**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация выпускника: Программист

Выполнила:

Студент 2-го курса группы ИСИП-22-4

Федорова И.М.

Проверил(а): Анатолий Николаевич Г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_преподаватель

Якутск 2024 г.

[**Тестовое Задание для Backend Разработчика** 3](#_Toc167110578)

[**Практическая часть** 4](#_Toc167110579)

# **Тестовое Задание для Backend Разработчика**

**Цель задания:** разработать веб-приложение для анализа и визуализации данных, собираемых с внешнего источника.

**Основные Требования:**

1. **Backend:**
   * Используйте **Laravel** для создания REST API.
   * Разработайте функционал для сбора данных из внешних источников. (data.csv)
   * Реализуйте обработку и агрегацию собранных данных.
   * Обеспечьте аутентификацию и авторизацию пользователей.
2. **Frontend:**
   * Используйте **Vue.js** для создания интерфейса, позволяющего пользователям взаимодействовать с данными.
   * Разработайте дашборд для визуализации данных с использованием таблиц.
   * Реализуйте фильтрацию, сортировку данных и настройку параметров отображения.
   * Обеспечьте доступ к дашборду только для авторизированных пользователей
3. **База данных:**
   * Используйте **MySQL** для хранения собранных данных и пользовательских настроек.
   * Оптимизируйте структуру базы данных для эффективной работы с большими объемами данных.

**Критерии оценки:**

* Качество кода и его организация (база данных).
* Производительность и масштабируемость решения.
* Качество пользовательского интерфейса и удобство в использовании.
* Полнота и качество документации проекта.

**Сроки и сдача работы:**

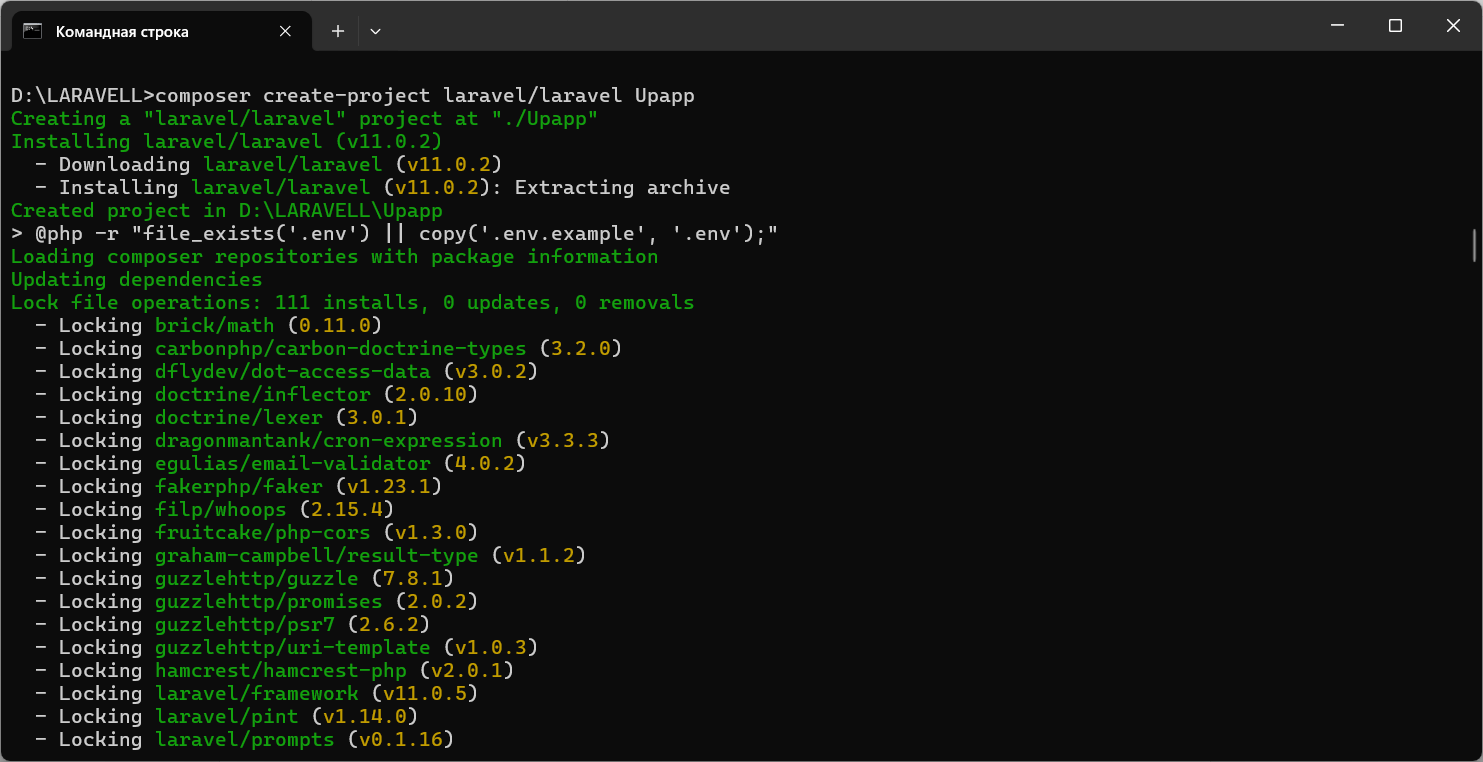
* Проект должен быть представлен в виде ссылки на Git-репозиторий, содержащий полный код проекта и инструкции для его запуска.

# **Практическая часть**

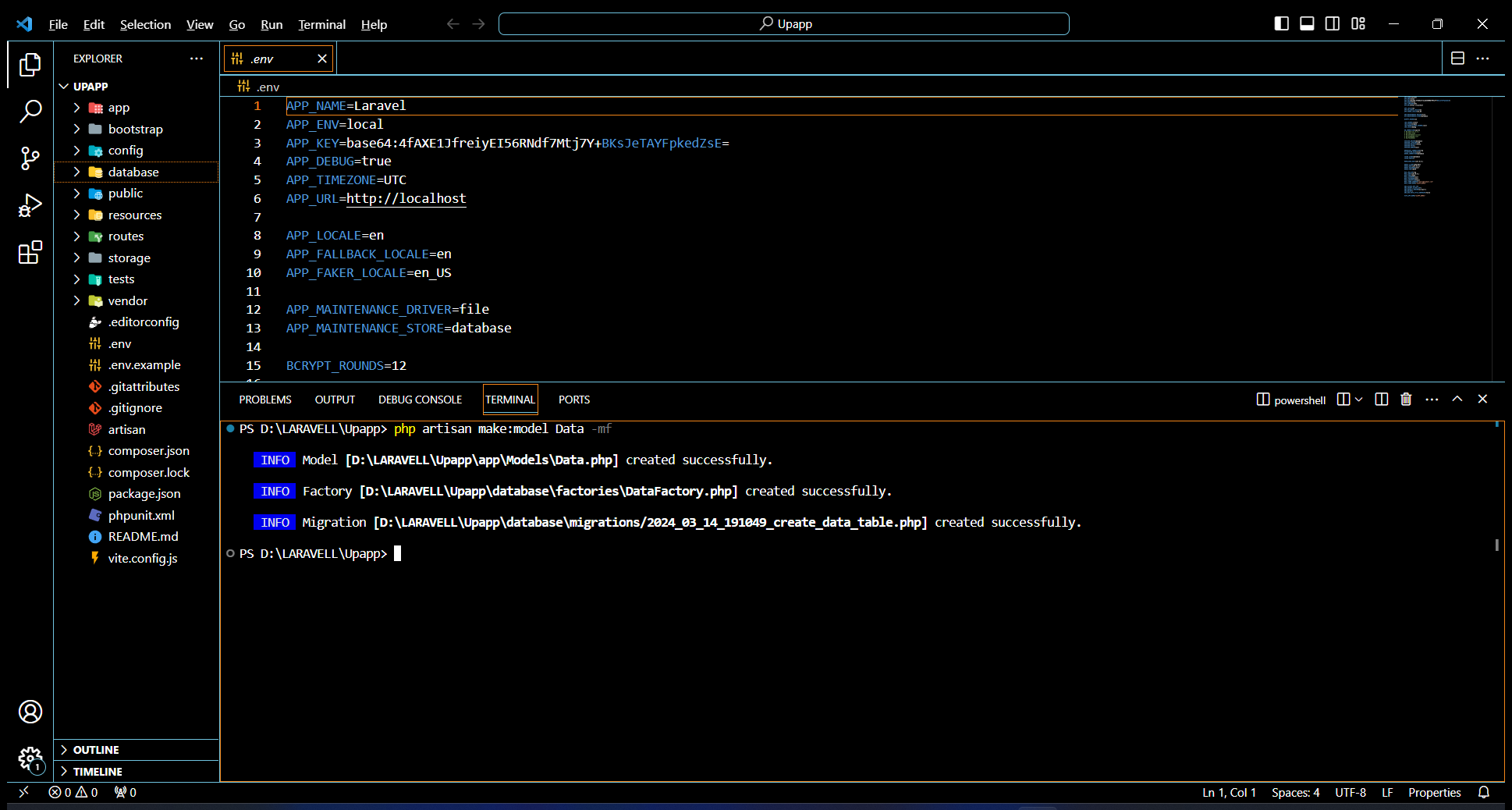
**Ход работы:**

1. **Backend:**

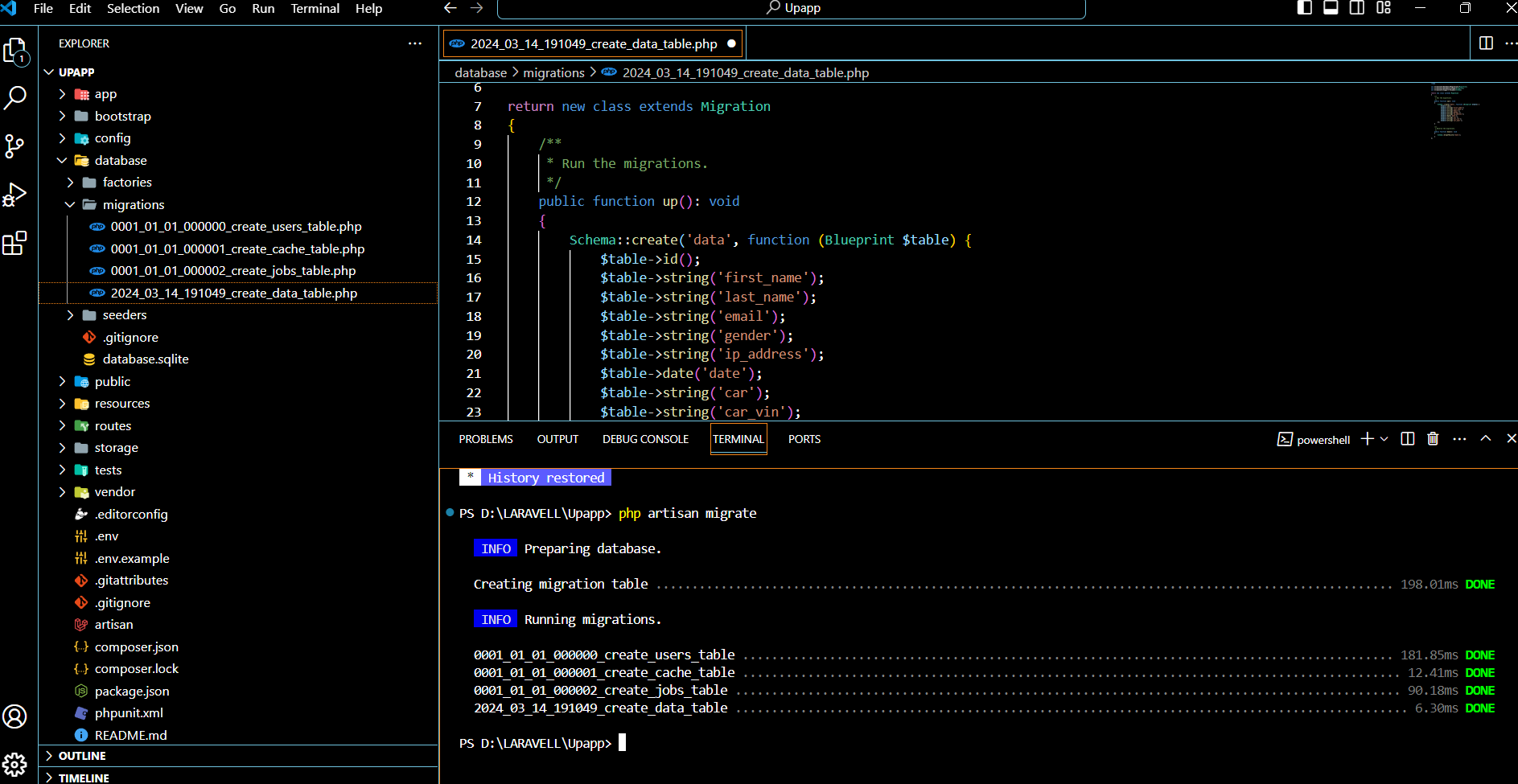
Во - первых установили и настроили laravel, composer и создали новый проект.



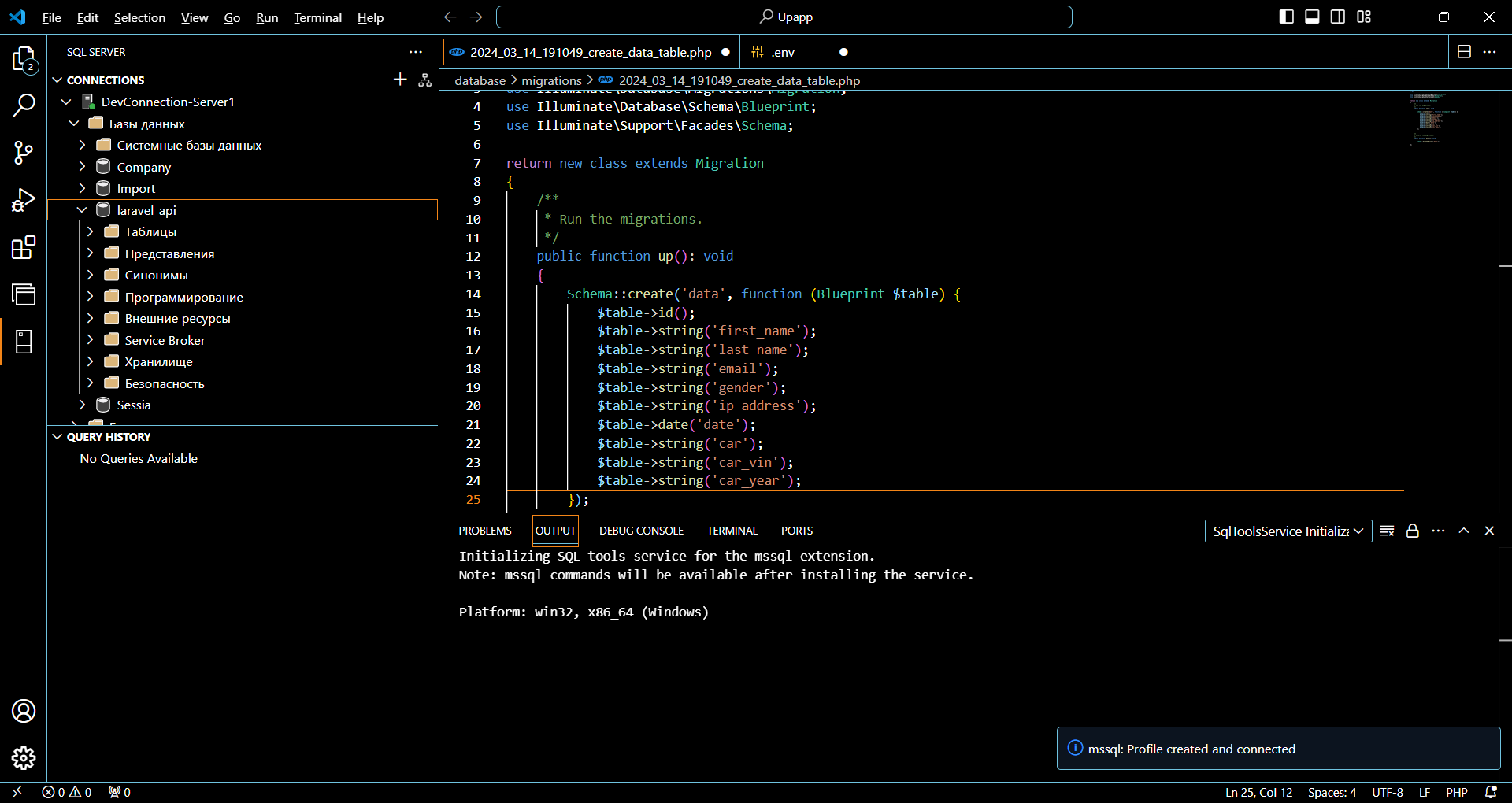
Создали модель Data



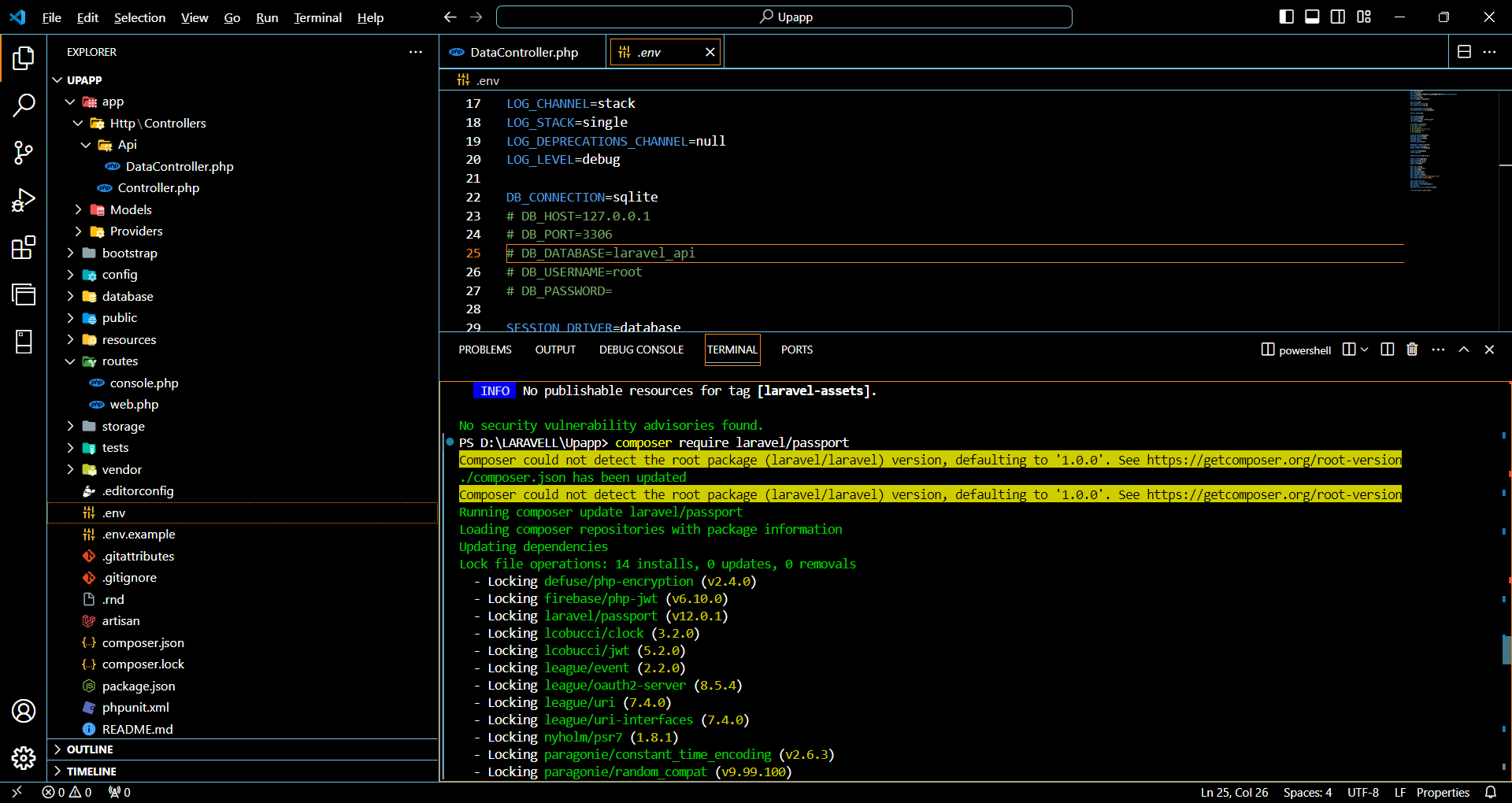
Далее создали миграцию в проекте



И соединили проект с базой данных



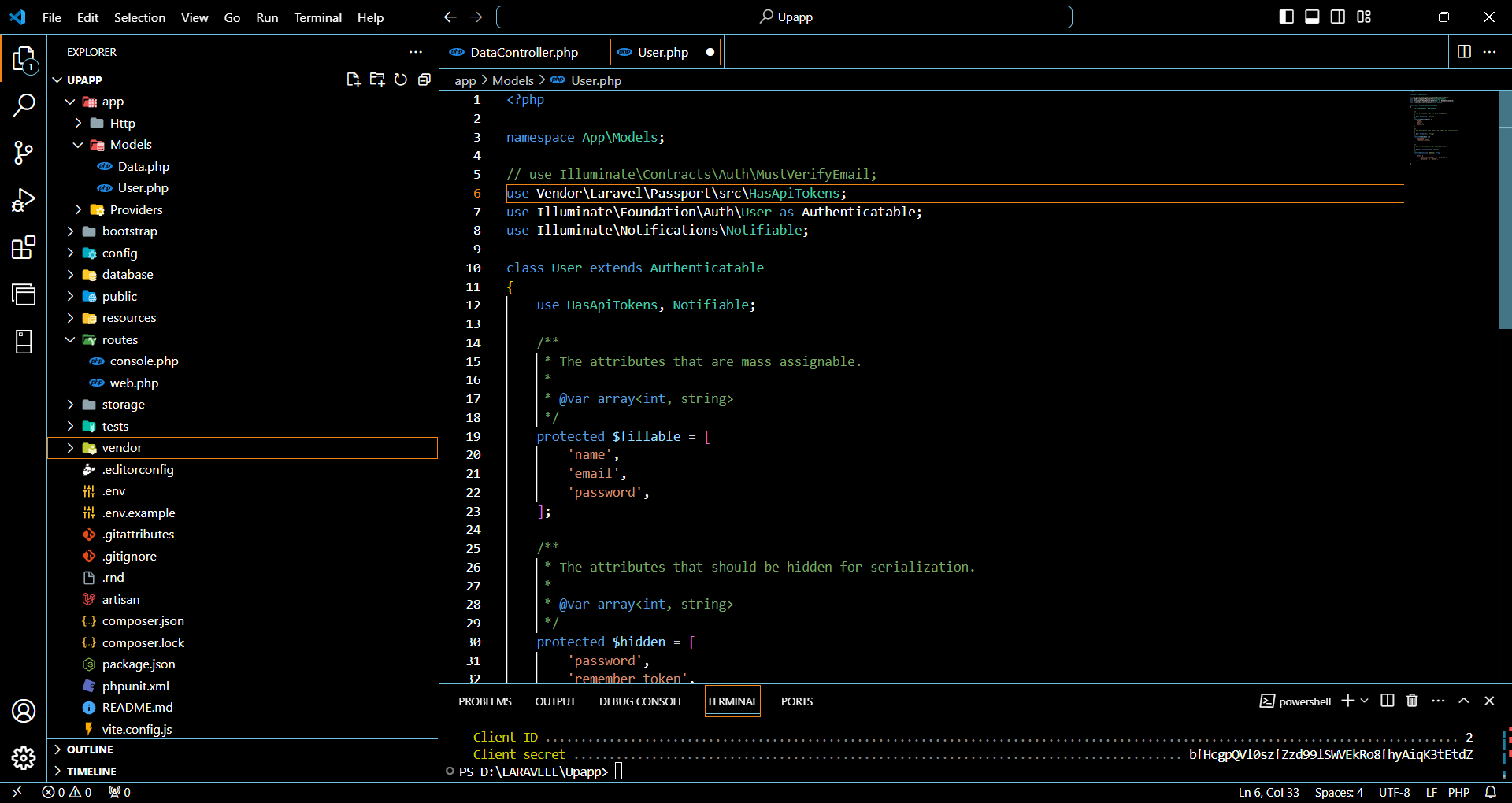
Установили laravel/passport



**Настройка паспорта**

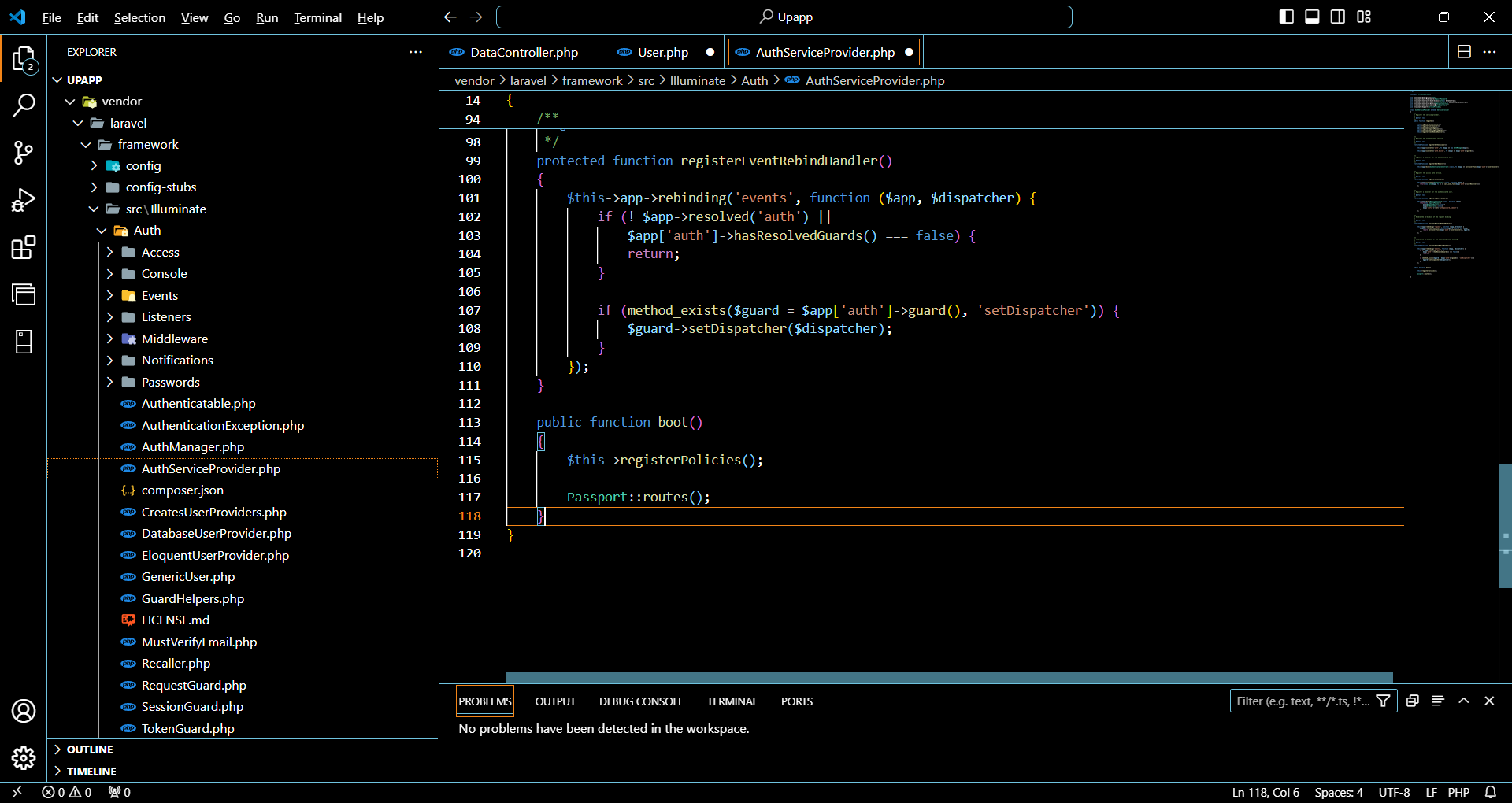
В модели добавили класс **HasApiTokens**,

**app/User.php**

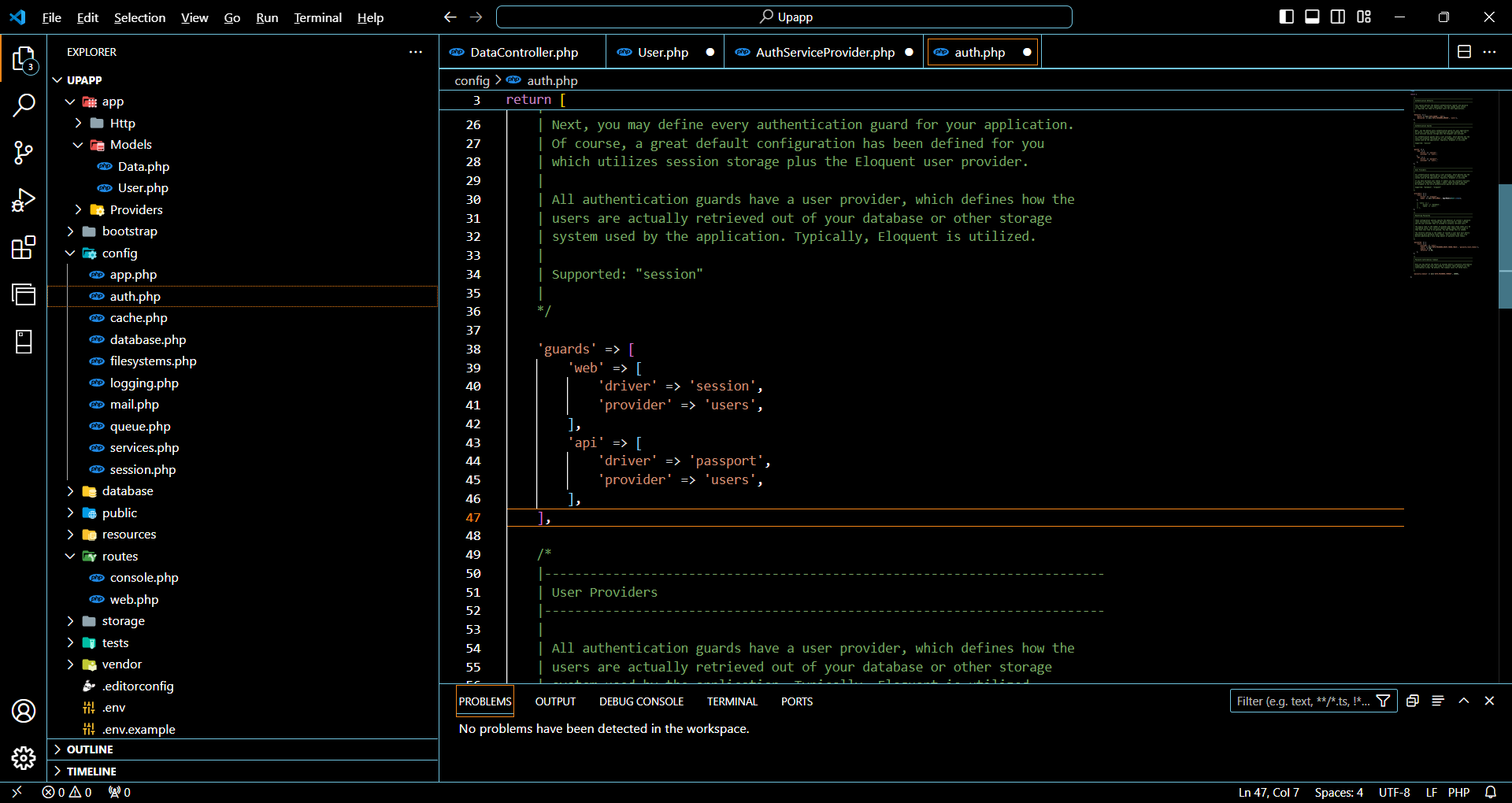


В **AuthServiceProvider** добавили **Passport::routes()**.

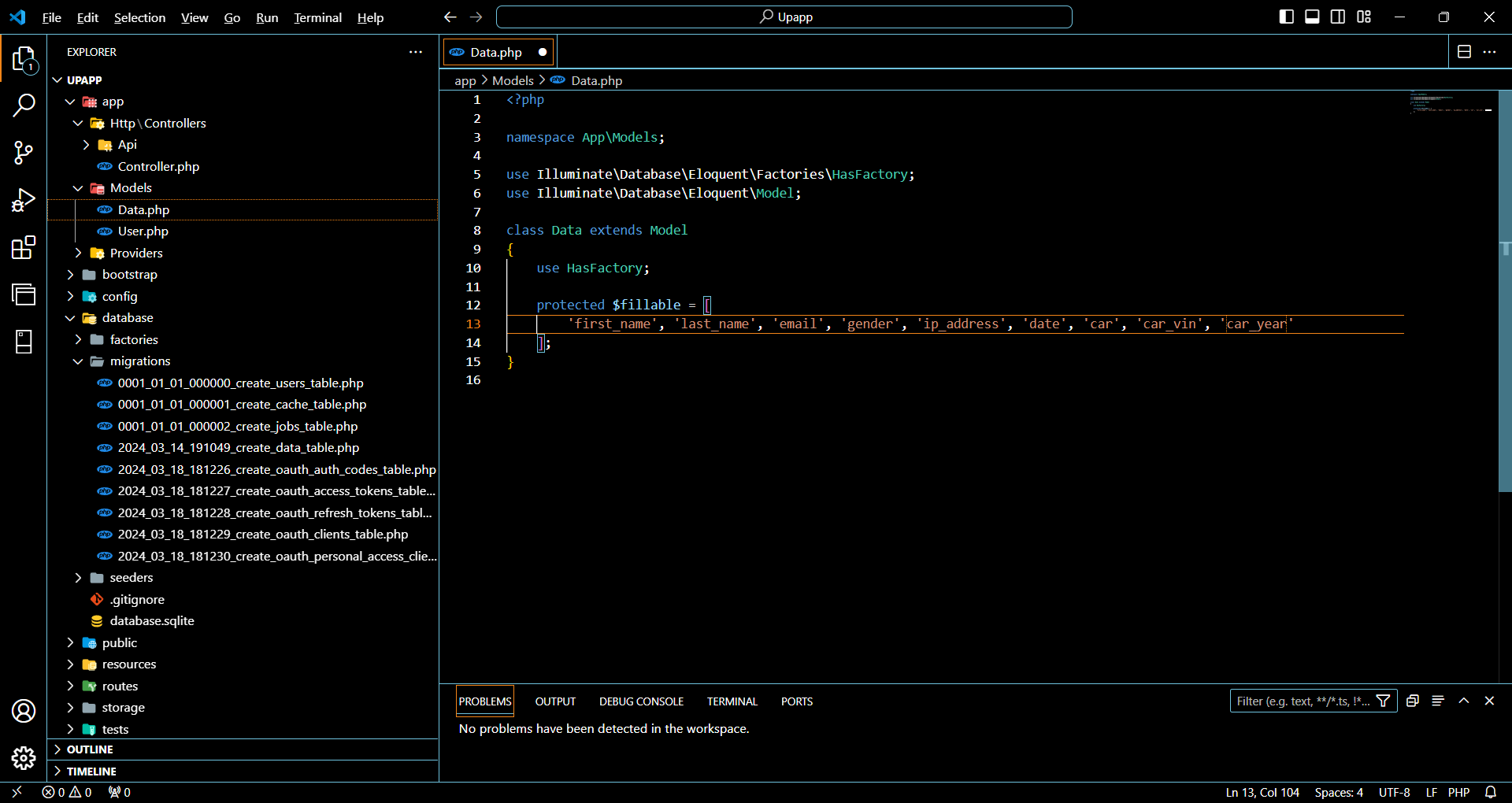
**app/Providers/AuthServiceProvider.php**



В файле **auth.php** добавили настройки аутентификации api.

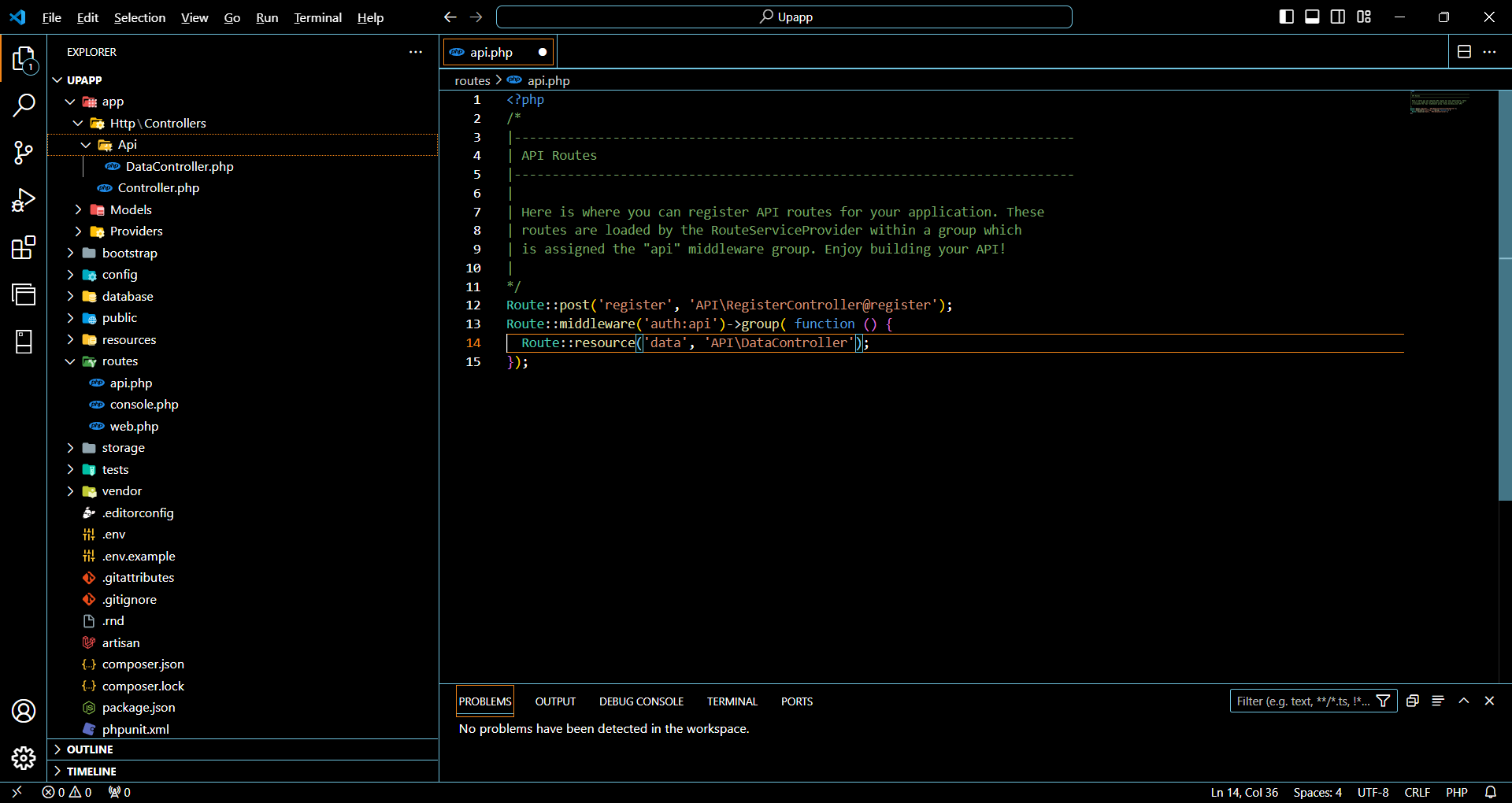


После настроили модель Data



**Создание маршрутов API**

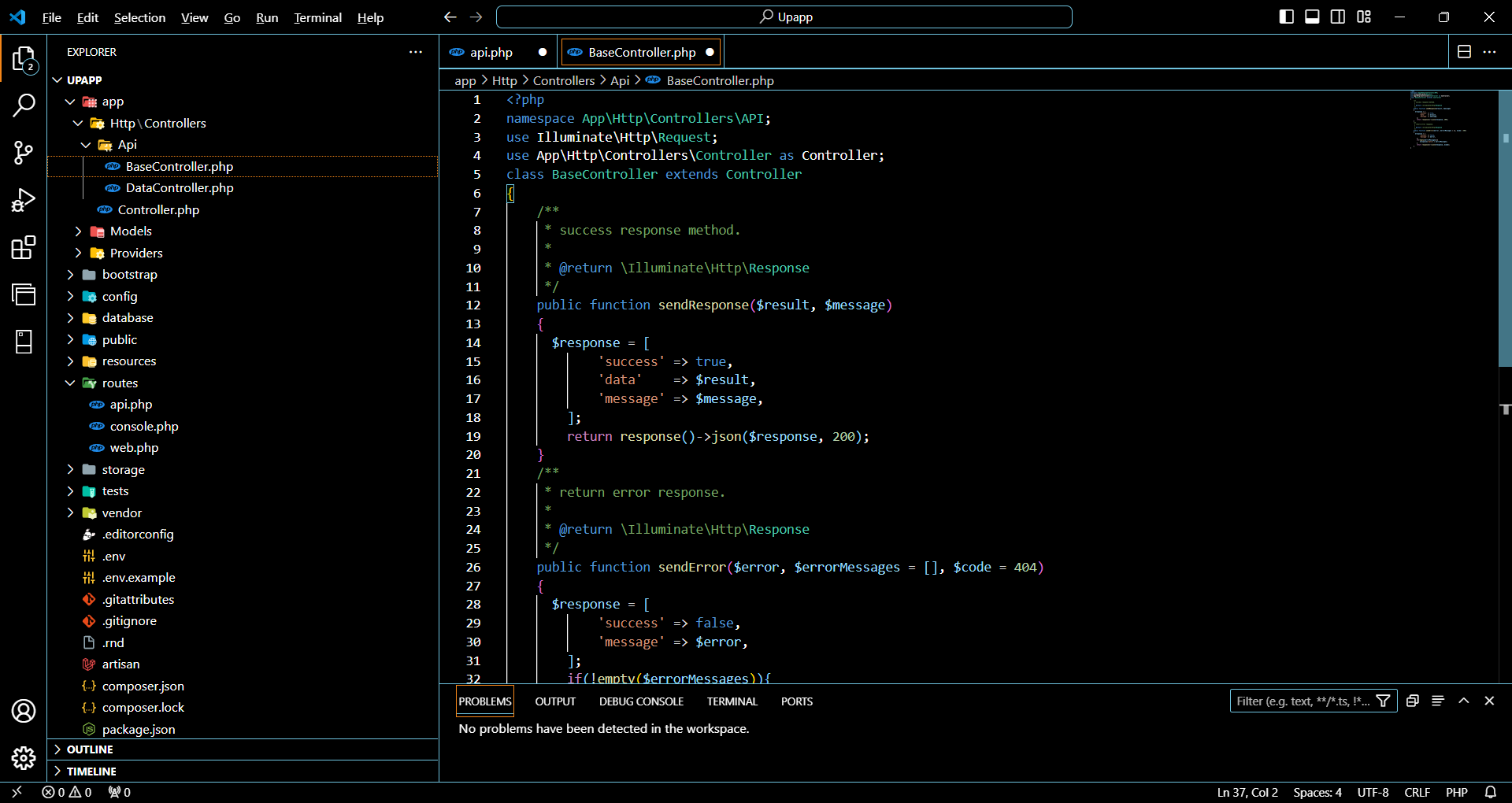
На этом этапе мы создали маршруты API.



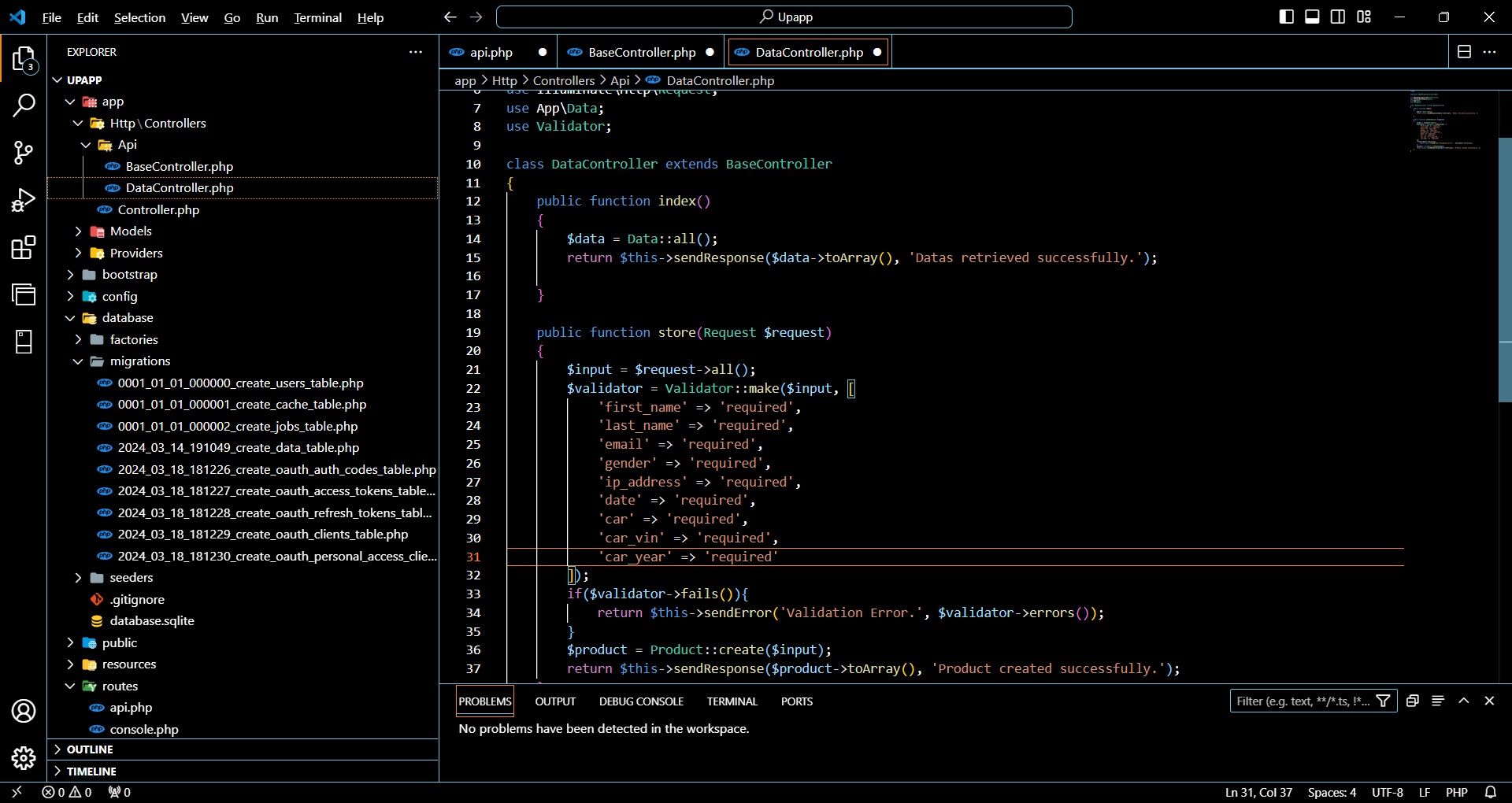
**Создание файлов Контроллера**

Здесь создали новые контроллеры.

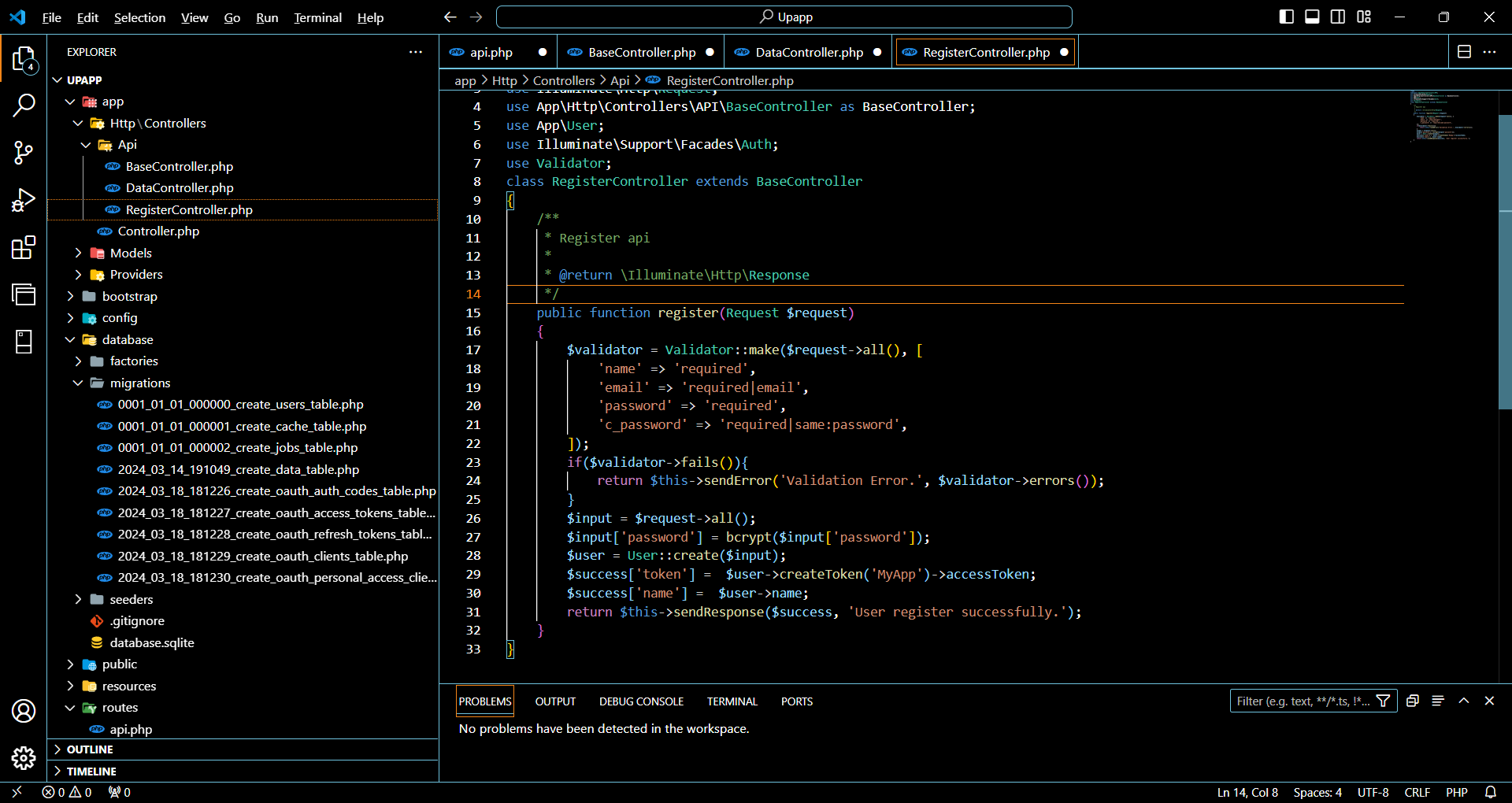
**app/Http/Controllers/API/BaseController.php**



**app/Http/Controllers/API/dataController.php**



**app/Http/Controllers/API/RegisterController.php**

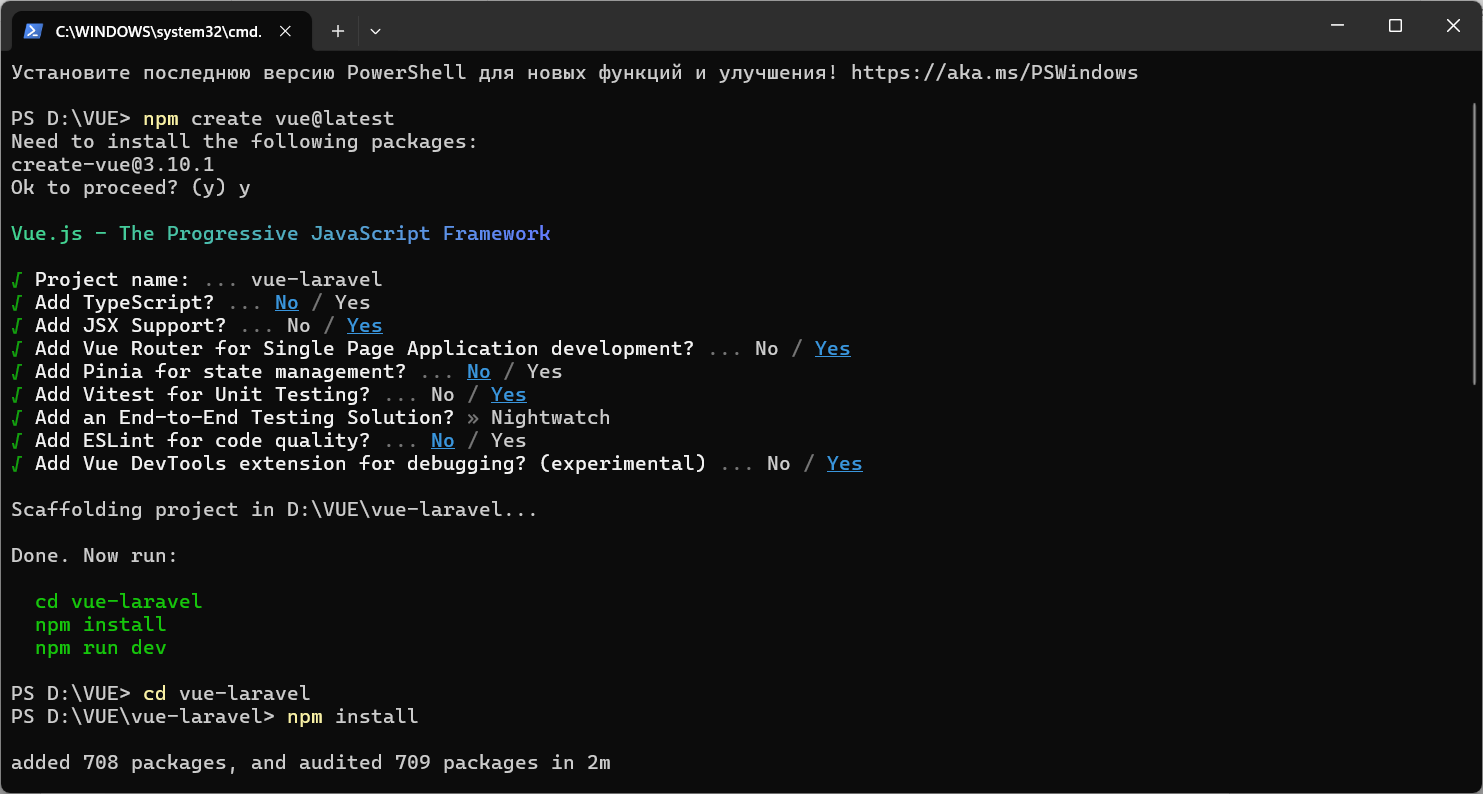


**Frontend:**

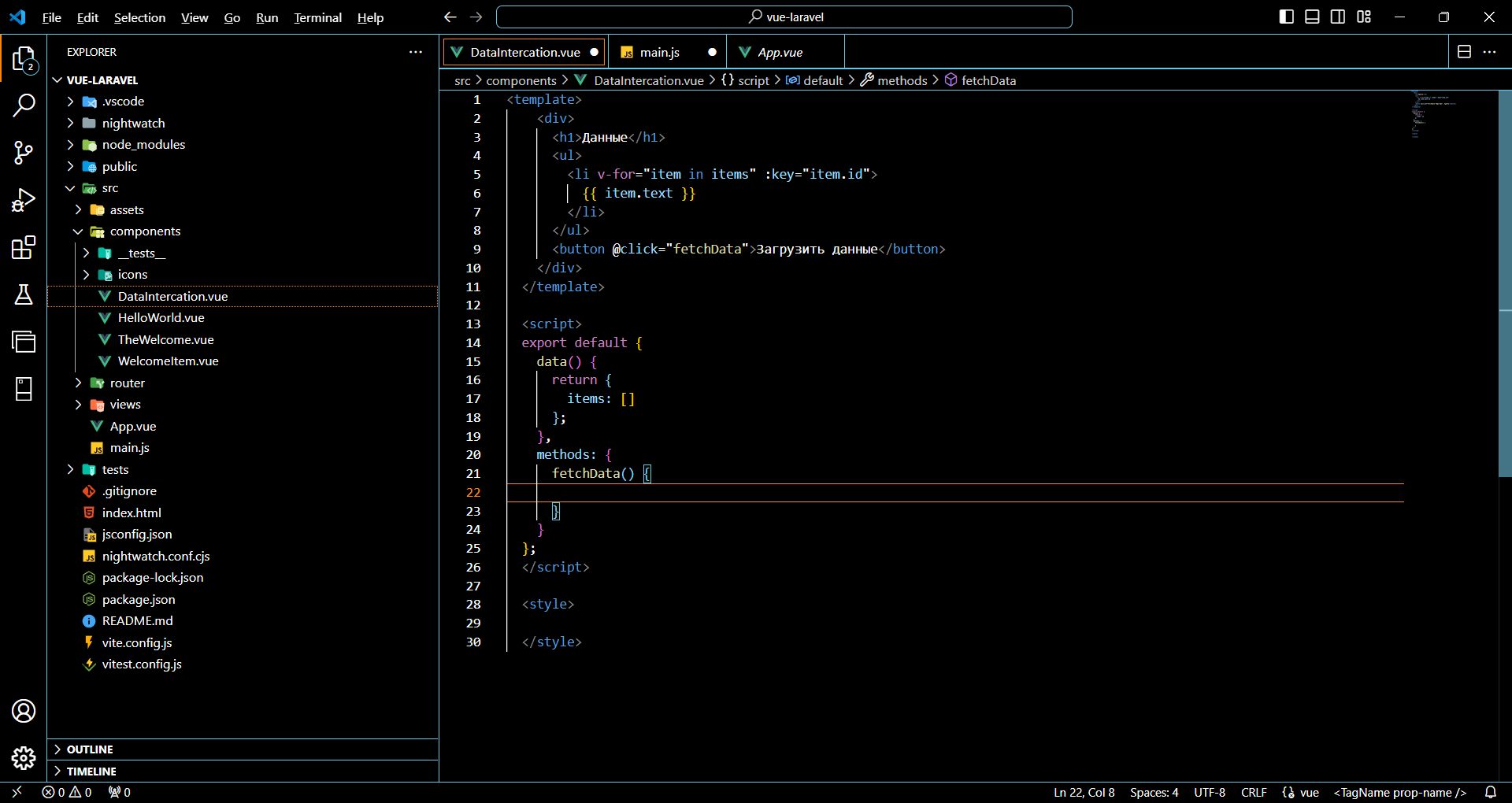
* + Используйте **Vue.js** для создания интерфейса, позволяющего пользователям взаимодействовать с данными.
  + Разработайте дашборд для визуализации данных с использованием таблиц.
  + Реализуйте фильтрацию, сортировку данных и настройку параметров отображения.
  + Обеспечьте доступ к дашборду только для авторизированных пользователей

Установили Node.js и npm на компьютере.

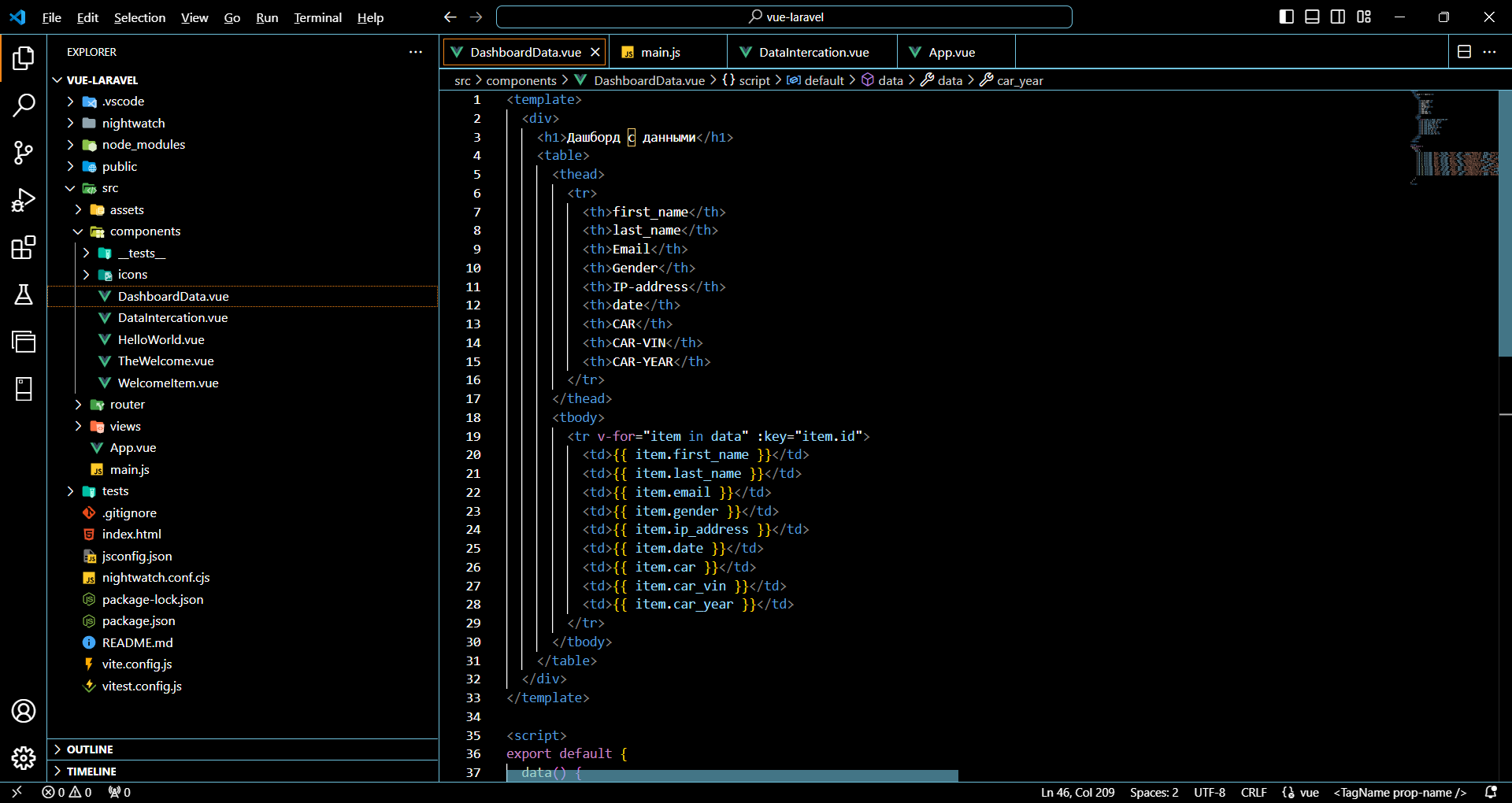
Создали проект на Vue.js



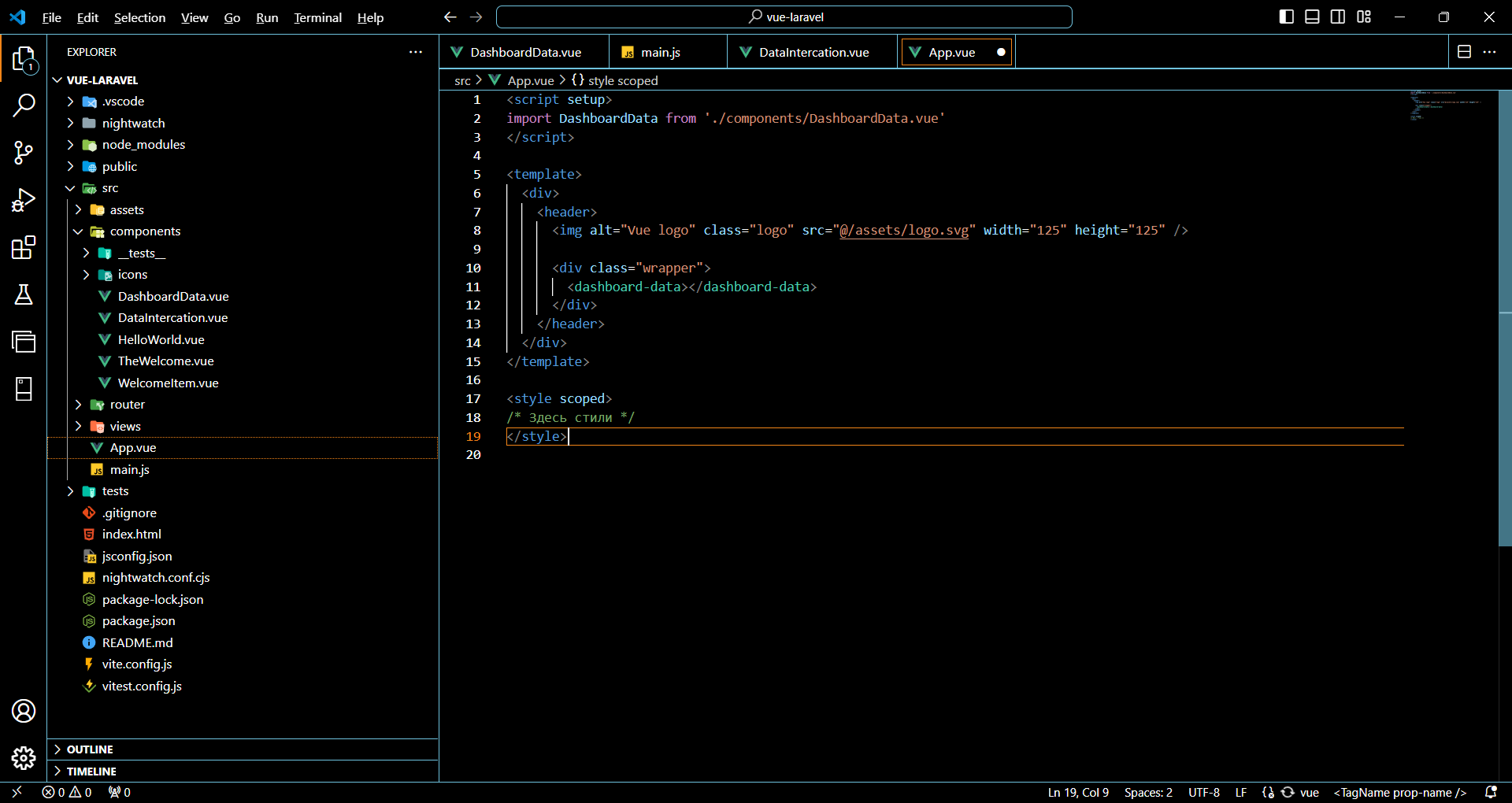
Создали итеративный файл в проекте datainteraction.vue и подключили его к главному файлу



Далее добавили файл dashboard.vue и прописали код.

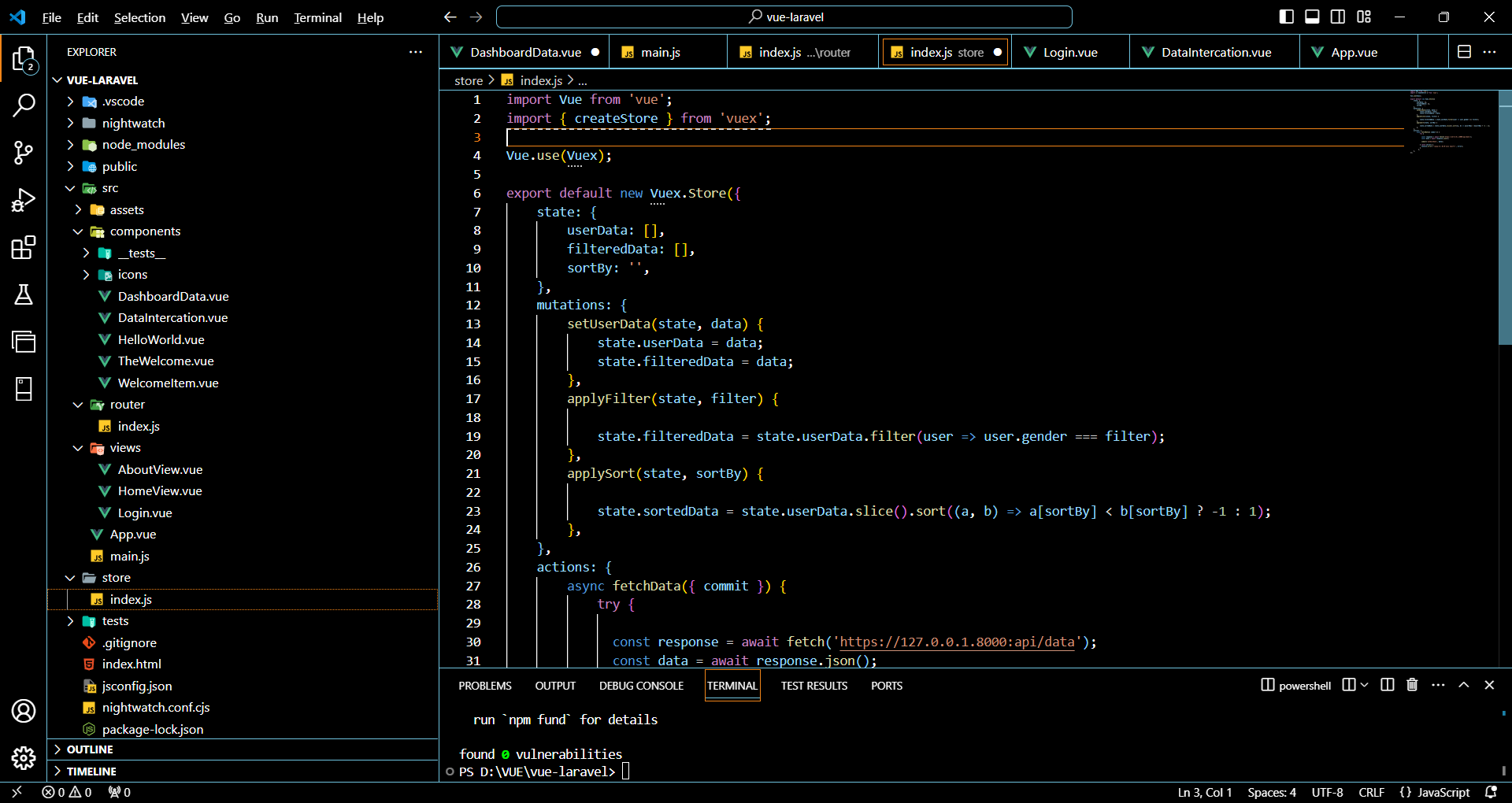


Изменили app.vue



Реализации фильтрации, сортировки данных и настройки параметров отображения

Создали папку store и файл index.js далее прописали код:



**База данных:**

Для базы данных создали БД laravel\_api в SQLSMS, где есть таблица с данными и пользователей.

