

Міністерство освіти і науки України
Шосткинський інститут сумського державного університету
Центр заочної та дистанційної форми навчання
Кафедра системотехніки та інформаційних технологій
Спеціальність 6.151 «автоматизація та комп'ютерно-інтегровані технології»

БАКАЛАВРСЬКА РОБОТА
на тему:

" Система автоматизованого управління ресторанним бізнесом "

Керівник роботи

викладач

П.С.Пата

Дипломник:

студент групи СУ-81Ш

В.В. Любивий

Номер залікової книжки:

Шостка – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ШОСТКИНСЬКИЙ ІНСТИТУТ СУМСЬКОГО ДЕРЖАВНОГО
УНІВЕРСИТЕТУ
КАФЕДРА СИСТЕМОТЕХНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Спеціальність 6.151.00.05 «Комп'ютеризовані системи управління та
робототехніка»

«ЗАТВЕРДЖУЮ»

Зав. кафедри системотехніки та
інформаційних технологій

_____ Г. М. Худолей

« 03 » травня 2022 р.

ЗАВДАННЯ

на бакалаврську роботу

студент **Любовий Вадим Вікторович**

1. Тема роботи Система автоматизованого
управління ресторанним бізнесом

Затверджено наказом директора інституту

№ 36 ОД від « 03 » травня 2022 р.

2. Строк здачі студентом закінченої роботи «21» червня 2022 р.

3. Вихідні дані до виконання роботи:

- завдання кафедри Розробити систему автоматизованого управління ресторанним бізнесом.

4. Зміст пояснювальної записки:

4.1 Аналіз складових частин ресторанного бізнесу як об'єкта управління.

4.2 Вибір типу програмної платформи.

4.3 Опис основних модулів.

4.4 Алгоритми роботи основних програмних рішень.

5. Перелік графічних матеріалів:

5.1. Функціональна схема бази даних системи.

7. Календарний план:

№ ета-пу	Зміст етапу роботи	Строк виконання (початок-кінець)
1	Аналіз завдання кафедри. Підбір та аналіз джерел інформації. Відбір аналогів та прототипів	08.05.2022 - 15.05.2022
2	Аналіз складових частин ресторанного бізнесу як об'єкта управління.	16.05.2022 - 29.05.2022
3	Вибір типу програмної платформи. Опис основних модулів. Алгоритми роботи основних програмних рішень	30.05.2022- 05.06.2022
4	Виконання програмної частини.	07.06.2022 - 12.06.2022
5	Технічне оформлення роботи. Здача роботи керівнику.	13.06.2022 - 20.06.2022

8. Дата видачі завдання 04 травня 2022 року

Керівник роботи ____ асистент, _____ Пата П.С.

вчені ступень та звання, посада

Підпис

Прізвище І.П.

Завдання до виконання прийняв:

Студент групи СУ - 81ш . _____ Любимий В.В.

Підпис

Прізвище І.П.

РЕФЕРАТ

Любивий Вадим Вікторович. Система автоматизованого управління ресторанним бізнесом. Бакалаврська робота. Шосткинський інститут Сумського державного університету. Шостка, 2022 рік.

Бакалаврська робота містить 57 аркушів пояснювальної записки, з урахуванням 26 рисунків, 1 таблиць; 2 креслень.

Роботу присвячено автоматизації ресторанного бізнесу. Вибрано і запрограмовано сучасну програмну платформу на мові PHP фреймворку Laravel. Розроблено основні модулі які враховують процеси всіх стейкхолдерів ресторанного бізнесу. Складені алгоритми та запрограмована система.

Ключові слова: система управління, фреймворк, канал Laravel, мова програмування PHP, програмна платформа.

SUMMARY

Lyubiviy V. V. Automated restaurant business management system. Bachelor work. Shostka Institute of Sumy State University. Shostka, 2022.

The bachelor's thesis contains 57 sheets of explanatory note, including 26 figures, 1 table; 2 drawings.

The work is devoted to the automation of the restaurant business. A modern software platform in the PHP language of the Laravel framework has been selected and programmed. The basic modules which consider processes of all stakeholders of restaurant business are developed. Compiled algorithms and programmed system.

Keywords: control system, framework, Laravel channel, PHP programming language, software platform.

Зміст:

- I. Вступ.
- II. Розділ I. Аналіз складових частин ресторанного бізнесу як об'єкта управління:
 - I. Вступ до розділу.
 - II. Аналіз переваг та недоліків різних систем автоматизації.
 - III. Аналіз ресторанного бізнесу з боку управління та обов'язків.
- III. Розділ II. Вибір типу програмної платформи:
 - I. Вступ до розділу.
 - II. Переваги та недоліки frameworks:
 - III. Порівняння фреймворків.
 - IV. Висновок з розділу.
- IV. Розділ III. Опис основних модулів:
 - I. Модульне програмування та структура:
 - II. Аналіз пакетів
 - III. Створення примітивного коду для аналізу принципів модульного проектування.
 - IV. Висновок з розділу.
- V. Розділ IV. Алгоритми роботи основних програмних рішень
- VI. Висновок
- VII. Список використаної літератури

Вступ:

Поява приватної власності у історії людства дала поштовх до розвитку нової соціальної складової – економіки, жага до збагачення активно розвивало її та створювало нові форми отримання доходів. Однією з таких форм стало поява Sobrino de Botín – першого у світі ресторану, який започаткував цілу гілку економічної науки.

Ресторанний бізнес – це особлива сфера діяльності, у якій найчастіше відбуваються зміни, зумовлені запитами та побажаннями гостей. Головна мета рестораторів – зробити щасливими тих людей, які прийшли до їхнього закладу. У більшості ресторанів працюють різні обслуговуючі співробітники, які подають їжу, напої та алкогольні напої. Але яким би прозорим з першого погляду не був ресторанний бізнес, він має багато проблем та низку своїх тонкощів.

Основна проблема – це робота людей з людьми. Кожна персона є індивідуальна: хтось легко обчислює великі числа, а хтось швидко забуває банальні речі. Але якими індивідуальними ми не були, у кожної людини є природні «вади»: ми маємо не бездоганну пам'ять, обмежені фізичні та інтелектуальні можливості, схильність до стресу через велике навантаження та банально втомлюємося через деякий період часу. Таким чином, через наші природні «вади» ми обмежені у виконанні роботи, бо одна людина не може готувати, прибирати та обслуговувати клієнтів у залі. Через це є місце розподілу праці. Таким чином з'являються прибиральниці, повари, офіціанти, адміністратори, тощо. Кожні з цих професій розподілені на більш конкретні, наприклад: помічник адміністратора, адміністратор та старший адміністратор (так є у великих ресторанах). Така велика диференціація сильно б'є по коштам власника, тому він вимушений шукати інші рішення. Одним з них є автоматизація.

Автоматизація — напрям науково-технічного прогресу, що використовує саморегулюючі технічні засоби та математичні методи з метою звільнення людини від участі в процесах отримання, перетворення, передачі та використання енергії, матеріалів, виробів або інформації, або суттєвого зменшення ступеня цієї участі або трудомісткості операцій, що виконуються. Автоматизація не тільки може замінити деякі професії взагалі, а також полегшити роботу персоналу. Автоматизація використовується дуже широко, починаючи від медичних операцій та закінчуючи магазинами без персоналу. Але ця робота стосується автоматизації у сфері ресторанного бізнесу. Дослідження проблеми автоматизації, пошуку систем автоматизації, виявити їх переваги та недоліки, вибору типу програмної платформи та опис різних алгоритмів – мета цієї роботи.

Розділ I. Аналіз складових частин ресторанного бізнесу як об'єкта управління

I. Вступ до розділу

Ефективне управління ресторанним бізнесом вимагає контролю над багатьма процесами, наприклад: управління цінами, знижками, доставкою, створення звітів, потрібної нам інформації у вигляді чистого прибутку та багато іншого. Ринок таких систем автоматизації дуже великий, тому важливо чітко розібрати їх переваги та недоліки.

II. Аналіз переваг та недоліків різних систем автоматизації.

1) Система автоматизації «Restik»:

1. Торгівля та склад:

Переваги:

- Можливість створювати замовлення,
- Управляти знижками, доставкою, цінами,
- Управління доступом до системи,

- Можливість пошуку та пошуку за фільтрами,
- Створення звітів та аналітика,
- Можливість прослідкувати переміщення товарів.

Недоліки:

- Відсутнє сканування штрих-кодів,
- Відсутній контроль заборгованості,
- Відсутній контроль за наявністю обов'язкового акцизу,
- Відсутнє резервування товарів,
- Відсутня інтеграція є маркетплейсами.

2. Автоматизація продажу:

Переваги:

- Моніторинг ефективності менеджерів,
- Перелік товарів та послуг,
- Підтримка характеристики товарів,
- Управління платежем,
- Система лояльності ,
- Офлайн-режим,
- Підключення фіскального реєстратора.

Недоліки:

- Відсутнє машинне прогнозування,
- Відсутній імпорт даних.

3. Онлайн-каси

Переваги:

- Товарний контроль,
- Продаж по вільній ціні,
- Звіти по продажам,
- Вивантаження даних,

- Офлайн-режим(робота з чеками без Інтернету),
- Пошук товарів,
- Управління торговими точками,
- Автоматичний підрахунок решти,
- Підключення принтеру,
- Можлива оплата: готівкова, безготівкова, змішана.

Недоліки:

- Відсутній сканер штрих-коду,
- Відсутнє вільне повернення.

4. Конструктор сайтів:

Переваги:

- Створення Інтернет-магазину,
- Система оплати онлайн,
- Особистий кабінет користувача,
- Резервне копіювання,
- База знань.

Недоліки:

- Відсутня Email-розсилка,
- Відсутня СМС-розсилка.

5. Безпека:

Переваги:

- Доступ до протоколу HTTPS,
- Резервне копіювання.

Недоліки:

- Відсутність багатофакторної авторизації.

Таким чином система автоматизації «Restik» має багато переваг і може слугувати повноцінною системою автоматизації ресторанного бізнесу. У системі є три продукти: автоматизація, сайт доставки та електронний менеджер QR. Також ця система має багато відгуків від користувачів. Якщо цю систему допрацювати, вона була б ефективніше, що можливо висунуло її на першу позицію ринку систем автоматизації.

2) Система автоматизації «iiko»

1. Торгівля та склад:

Переваги:

- Управління цінами,
- Управління знижками,
- Облік фінансів,
- Контроль заборгованості,
- Сканування штрих –кодів,
- Пошук / фільтри,
- Управління доставкою,
- Стосунки з постачальниками,
- Управління рухом товарів,
- Звіти та аналітика,
- Контроль доступу,
- Підтримка властивостей товарів.

Недоліки

- Відсутнє резервування товарів,
- Відсутні шаблони форм.

2. Автоматизація продажу:

Переваги:

- Список товарів та послуг,
- Управління цінами,

- Управління знижками,
- Управління платежів,
- Виставлення рахунків,
- Імпорт даних,
- Звіти,
- Режим офлайн,
- Підключення фіскального реєстратора,
- Системи лояльності,
- Підтримка властивостей товарів.

Недоліки:

- Відсутнє машинне прогнозування,
- Відсутній моніторинг менеджерів.

3. Онлайн-каси:

Переваги:

- Режим офлайн,
- Товарний облік,
- Вивантаження даних,
- Управління роздрібними торговими точками,
- Продаж за безкоштовною ціною,
- Звіти про продажі,
- Автоматичний розрахунок доставки,
- Повернення товарів,
- Грошові зміни,
- Затримані чеки,
- Розрахунок: готівка, безготівка.

Недоліки:

- Відсутність пошуку товарів.

4. Програми лояльності:

Наявність:

- Електронна картка знижок,
- Шаблони програми знижок,
- Багаторівневі програми,
- Критерії програм,
- Платні подарункові картки,
- Інтеграція з сайтом,
- Гейміфікація,
- SMS/електронна пошта,
- Групи клієнтів,
- Нагороди за дії.

5. Безпека:

Переваги:

- Доступ до протоколу HTTPS,
- Резервне копіювання в декількох місцях.

Недоліки:

- Відсутність багатофакторної авторизація.

Система автоматизації «iiko» має багато переваг, але немає доступу до створення сайту, якщо порівнювати з «Restik». Але саме «iiko» передбачає велику програму лояльності, що у будь-якому разі привалює клієнтів. Iiko спрощує роботу з касиром, складом, кухнею, фінансами, гостями, персоналом та постачальниками.

Крім того, в ІКО ви можете знайти управління доставкою готових страв, бухгалтерського обліку та моніторингу випуску готових страв, управління виконанням замовлень, моніторинг часу підготовки та подання страв, інформування гостей про готовність. Можливість замовити за допомогою табло, система Von-Deposit та інші параметри. Я вважаю система «iiko» має великі перспективи у розвитку.

3) Система автоматизації «Poster»

1. Автоматизація продаж:

Переваги

- Моніторинг ефективності менеджерів,
- Список товарів та послуг,
- Управління цінами,
- Управління знижками,
- Управління платежами,
- Імпорт даних,
- Звіти,
- Режим офлайн,
- Підключення фіскального реєстратора,
- Системи лояльності,
- Підтримка властивостей товарів.

Недоліки:

- Відсутність машинного прогнозування.

2. Взаємодія з клієнтами:

Переваги:

- Експорт/імпорт даних,
- Зберігання файлів,
- Технічна підтримка,
- Електронні бюлетені електронної пошти,
- Маркетингові інструменти,
- Завдання та графік,
- Управління клієнтами/партнерством,
- Кількість користувачів: необмежена кількість,
- Працювати з транзакціями та оплатою,
- Внутрішній пошук/фільтри.

Недоліки:

- Інтеграція з IP-телефону/Coll-Center,
- Немає роботи з документами,
- Немає більше воронки продажів,
- Немає історії взаємодії з клієнтом,
- Налаштування поля немає,
- Немає категорії потенційних клієнтів.

3. Безпека:

Переваги:

- Доступ до протоколу HTTPS,
- Резервне копіювання в декількох місцях.

Недоліки:

- Відсутня багатфакторна авторизація.

Система автоматизації складається з фронт-офісу для працівників залу та підтримки адміністративної панелі для менеджерів та власників. Також є спеціальна заявка для власників - Boss Poster, де в режимі реального часу відображаються основні звіти: дохід, прибуток, кількість чеків, середній час обслуговування тощо.

Передній офіс системи працює на планшетах Android, Windows, iPad та ноутбуках, які можуть заощадити на обладнанні під час відкриття закладу.

Основні функції фронтального офісу:

- Швидке навчання персоналу.
- Надійна робота в режимі офлайн.
- Режим Fastfood або карта залу.
- Друк перевірки, бігуни на кухні.
- Змішані методи оплати.
- Бронювання столу

- Зміни працівників працівників
- Інтернет -замовлення
- Робота з фіскальним реєстратором.
- Бухгалтерський облік грошей при оформленні каси.
- Знижки, бонуси, акції.
- Клієнтська картка.

Основні функції заднього офісу:

- Побудовані та налаштовані звіти.
- Фінансовий та складський облік.
- Налаштування доступу.
- Системи лояльності.
- Історія дій для кожної перевірки.
- Купівля нагадування.
- Напівфінансовані продукти, пакети, інвентар.
- Управління мережею.
- Управління франшизами.
- Облік зарплати
- Аналітика продажів
- Налаштування безпеки.

Я бачу систему «Poster» більш цікавою у автоматизації ресторанного бізнесу через низку її переваг. Також система вийшла з ринку РФ та Білорусі, що також є фактором вибору системи для певних соціальних груп або окремих персон.

4) Система автоматизації «Quick Resto»

1. Торгівля та склад:

Переваги:

- Є управління цінами,
- Є управління знижками,
- Є сканування штрих –кодів,

- Є інвентар,
- Є пошук / фільтри,
- Є рух товарів,
- Є звіти та аналітика,
- Є контроль доступу,
- Є шаблони форми,
- Існує підтримка властивостей товарів,
- Є виробничі замовлення,
- Є технологічні карти.

Недоліки:

- Немає бухгалтерського обліку фінансів,
- Немає контролю заборгованості,
- Немає управління доставкою,
- Немає стосунків з постачальниками,
- Немає застереження товарів,
- Немає цільового зберігання.

2. Автоматизація продаж:

Переваги:

- Існує моніторинг ефективності менеджерів
- Існує список товарів і послуг
- Є управління цінами
- Є управління знижками
- Є управління платежами
- Є рахунок -фактура
- Є імпорт даних
- Є звіти,
- Є режим офлайн,
- Існує зв'язок фіскального реєстратора,
- Є системи лояльності,

- Існує підтримка властивостей товарів.

Недоліки:

- Відсутнє машинне програмування.

3. Взаємодія з клієнтами:

Переваги:

- Експорт/імпорт даних,
- Є магазин файлів,
- Є технічна підтримка,
- Є маркетингові інструменти,
- Існує управління базою даних клієнтів/партнерів,
- Робота з документами,
- Робота з транзакціями та платежами,
- Є внутрішній пошук/фільтри.

Недоліки:

- Немає розсилки електронної пошти,
- Немає завдання та розкладу,
- Немає інтеграції з IP -телефонією/кол-центром,
- Немає історії взаємодії з клієнтом,
- Налаштування поля немає,
- Немає категорії потенційних клієнтів.

4. Безпека:

Переваги:

- Доступ до протоколу HTTPS,
- Є багатофакторна авторизація,
- У кількох місцях є резервне копіювання.

Недоліків немає.

Система складається з фронтового офісу та підкладки з системою автономної експлуатації, структури CRM, складів та механізмів відстеження персоналу.

Переваги фронтового офісу:

- Чіткий і простий інтерфейс, без необхідності навчання персоналу.
- Доступність - не потрібно купувати сервери та стежити за ними. Дані зберігаються у хмарі.
- Автономна робота - термінали доступні навіть за відсутності мережі. Продажі та дії персоналу будуть синхронізовані, якщо це необхідно, та поява Інтернету.
- Підтримка стороннього обладнання, включаючи фіскальні реєстратори, принтер квитків, сканери та штрих-коди, термінали тощо.

Особливості Back Office:

- Налаштування прав доступу, встановлення дозволів та заборони на конкретні дії персоналу.
- Зручні термінали: прийняття замовлення, відправлення його на кухню та оплату трьома діями.
- Підтримка різних видів платежів.

Система автоматизації також пропонує CRM:

- Різні програми лояльності - залучення нових клієнтів та підтримують постійні.
- Вибір типів карт - це параметри з кодом, магнітною смужкою або безконтактною.
- Статистика з даними про затверджені страви, середня та загальна кількість замовлень тощо.
- Складовий облік з розрахунком, контроль над рухом товарів та звітів.

Я вважаю ця система автоматизації може стати передовою на ринку, якщо її доповнити деталями, які будуть притаманні тільки їй. Взагалі система дуже цікава на мій погляд.

5) Система автоматизації «Dooglys»

1. Торгівля та склад:

Переваги:

- Створення замовлень,
- Є управління цінами,
- Є управління знижками,
- Є облік фінансів,
- Є сканування штрих –кодів,
- Є інвентар,
- Є пошук / фільтри,
- Є управління доставкою,
- Є стосунки з постачальниками,
- Є рух товарів,
- Є звіти та аналітика,
- Є контроль доступу,
- Є шаблони форми,
- Існує підтримка властивостей товарів,
- Є цільове зберігання,
- Є виробничі замовлення,
- Є технологічні карти.

Недоліки:

- Немає інтеграції з ринками.

2. Автоматизація продаж:

Переваги:

- Є прогнози,
- Існує список товарів і послуг,
- Є управління цінами,

- Є управління знижками,
- Є управління платежами,
- Є імпорт даних,
- Є звіти,
- Є режим офлайн,
- Існує зв'язок фіскального реєстратора,
- Є системи лояльності,
- Існує підтримка властивостей товарів,
- Існує створення сценаріїв продажів,
- Існує автоматизація звичайних операцій.

Недоліки:

- Немає моніторингу ефективності менеджерів.

3. Програма лояльності:

Переваги:

- Є електронна картка знижок,
- Є програми знижок,
- Є багаторівневі програми,
- Є програми програм,
- Є платні подарункові картки,
- Існує інтеграція з сайтом,
- Є SMS/електронна пошта,
- Є групи клієнтів,
- Є нагороди за дії.

Недоліки:

- Гейміфікації немає.

4. Онлайн-каси:

Переваги:

- Є режим офлайн,
- Є товарний облік,
- Є вивантаження даних,
- Існує управління торговими балами,
- Є продажі за вільною ціною,
- Є пошук товарів,
- Є звіти про продаж,
- Існує автоматичний обчислення доставки,
- Є повернення товарів,
- Є зміни касових реєстру,
- Є перенесені чеки,
- Розрахунок: готівка, картка,
- Існує інтеграція з банківськими терміналами,
- Існує з'єднання сканера штрих –коду,
- Існує з'єднання принтера мітки,
- На одній каси є кілька систем оподаткування.

Недоліків немає.

5. Безпека:

Переваги:

- Доступ до протоколу HTTPS,
- Є багатофакторна авторизація,
- У кількох місцях є резервне копіювання.

Недоліків немає.

Система «Dooglys» має дуже багато переваг у порівнянні з недоліками. Поміж усіх систем, які було розглянуто, ця система найкраща та найпрогресивніша.

б) Система автоматизації «POS FUSION»

1. Автоматизація продаж:

Переваги:

- Існує моніторинг ефективності менеджерів,
- Існує список товарів і послуг,
- Є управління цінами,
- Є управління знижками,
- Є звіти,
- Є режим офлайн,
- Існує зв'язок фіскального реєстратора,
- Є системи лояльності,
- Існує підтримка властивостей товарів.

Недоліки:

- Прогнозу немає,
- Немає управління платежами,
- Немає рахунків,
- Немає імпорту даних.

2. Програма лояльності:

Переваги:

- Є електронна картка знижок,
- Є програми знижок,
- Є багаторівневі програми,
- Є групи клієнтів.

Недоліки:

- Програм немає,
- Немає платних подарункових карток,
- Немає інтеграції з сайтом,
- Гейміфікації немає,
- Немає інформаційних бюлетенів SMS/електронної пошти.

3. Безпека:

Переваг немає.

Недоліки:

- Немає доступу до протоколу HTTPS,
- Немає багатофакторної авторизації,
- Немає резервного копіювання в декількох місцях.

4. Можливості обслуговування:

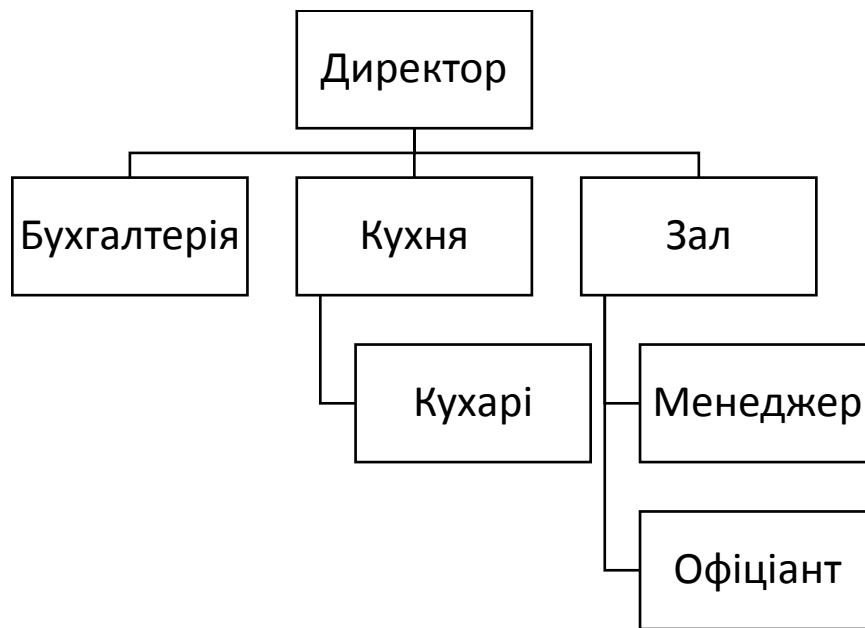
- Управління меню.
- Управління товарами, стравами, напівфінансованими продуктами.
- Технологічні карти.
- Управління складом.
- Огляд виробництва та запасів.
- Помилковий.
- Система бухгалтерського обліку спочатку в, спочатку вийти
- Лояльне управління: знижки, клієнти, депозити, бонуси.
- Графіки, звіти, аналітика.

III. Аналіз ресторанного бізнесу з боку управління та обов'язків.

Велике розмаїття програм кажуть нам про те, що це користується попитом. Попит є показником інтересу з боку користувача. Це справді зручно, аналізуючи ці програми я дійшов висновку, що робота кожного з персоналу стає набагато легше.

Хочу порівняти історію управління ресторанним бізнесом до розвитку автоматизації та після та розглянути більш детально структуру ресторанного бізнесу.

Зробимо схему простого ресторану:



1. Бухгалтерія.

Бухгалтерія є джерелом документально обґрунтованої та структурованої економічної інформації, необхідної для прийняття управлінських рішень з метою забезпечення ефективного господарювання. Бухгалтер працює з числами, зайнята обробкою даних, створенню звітів та документальним оформленням. Це потребує спеціальних навичок, тобто спеціаліста, які зазвичай потребують не малої заробітної платні. Якщо раніше бухгалтер повинен був все робити власноруч або за допомоги примітивних програм, то у добу автоматизації ці функції виконує програма. Переваги програми над людиною зрозумілі. Якщо людина може помилитися, а у роботі бухгалтера це грає дуже велику роль, бо впливає на подальший розвиток бізнесу, то програма, при правильному написанні, не робить помилок, працює швидше та ефективніше. Таким чином автоматизація у майбутньому може замінити бухгалтера, що набагато знизить економічні витрати підприємства та повисить чистий прибуток директорові, або створе поле грошей, які можна буде використовувати на благо бізнесу. Ці аргументи показують нам як саме автоматизація допомагає підприємству ці сторони автоматизації бухгалтерської роботи.

2. Кухар.

У обов'язки кухаря входить приготування. Він тісно пов'язаний з менеджером, який стоїть на касі, бо той повинен проінформувати про те, що саме і у якій кількості треба приготувати. За часи до автоматизації були паперові листи, де була вказана вся потрібна інформація. У особливо важкі дні, коли клієнтів забагато, це було дуже незручно: могли помилитися з кількістю, стравою або індивідуальними запитами клієнтів. Автоматизація допомогла у цій проблемі. Тепер кухарю приходиться все на пристрій, де чітко вказана вся потрібна інформація. Також автоматизація показує інформацію про приготування: яка кількість, які способи, тощо. Для професійного кухаря це не значний додаток, але все ж таки це дуже зручно та корисно. Таким чином автоматизація позбавила людей незручного способу інформування між менеджером та кухарем.

3. Менеджер.

Менеджер стоїть за касовим апаратом. Він повинен приймати замовлення, видавати решту, обслуговувати за системою лояльності та повідомляти кухню щодо замовлення. Раніше потрібно було все записувати на папері та передавати, рахувати решту за допомоги калькуляторів. Це звісно ж не дуже зручно, до того ж використовує додатковий матеріал. Автоматизація процесу допомогла менеджеріві, бо зараз замість ручки у нього стилус, а папір замінили екраном планшета або іншого зручного пристрою. Не потрібно передавати інформацію, коли її можна надіслати на кухню усього за один – два кліки. Автоматизований процес розрахунку решти, коли програма сама усе розраховує більш точно, бо людина завжди може помилитися та розрахувати її не так, за що потім може настати відповідальність перед директором. Це дуже зручно та більш ефективно, бо час обслуговування однієї людини стає набагато менше.

4. Офіціант.

Якщо взяти стандартну модель ресторану, то роль менеджера також виконує офіціант. Окрім цього офіціант повинен записувати замовлення, передавати його менеджеріві або кухні, залежить від наявності першого. Також офіціант

повинен запам'ятати до якого столу яка страва. Це звісно складно запам'ятати, особливо у складні дні. Автоматизація створює візуалізацію, яка чітко показує офіціантові яку страву до якого столу. Також ці програми можуть допомагати офіціантові у пропозиції та рекламі тих чи інших страв. Таким чином автоматизація теж полегшує роботу офіціантові.

5. Директор.

Директор зазвичай обирає курс розвитку підприємства, рахує або аналізує розрахунки бухгалтера з приводу прибутку та витрат. Раніше бути вправним директором означало бути досвідченим у економіці, математиці та просто мати підприємницькі вміння. Але зараз процес автоматизації допомагає у машинному передбаченні, прогнозуванні, побудові графіків, поданні звітів і багато іншого. Зараз бути директором не завжди означає бути вправним економістом, тепер можна просто вміти аналізувати дані та знаходити перспективи розвитку. У будь-якому разі робота директора стала значно легше.

IV. Висновок з розділу.

У цьому розділі ми розглянули шість систем автоматизації ресторанного бізнесу. Деякі системи мають переваги у певній сфері, інші – в іншій, але об'єктивно можна говорити о «поганих» та «добрих» системах. За функціоналом краще за усіх показав себе «Dooglys». Великий спектр можливостей цієї системи дозволяє контролювати бізнес з усіх боків. Недоліків мало, на них можна не звертати увагу, бо вони не значні. Саме ця система задовольняє усі критерії оцінювання: від комунікації з клієнтами до безпеки. Я вважаю цю систему найкращою з тих, про які я чув. Задля того, щоб показати реальну різницю між функціонуванням ресторану до і після була створена примітивна схема, яка підходить будь-якому ресторанові, за якою були пояснені та аргументовані переваги автоматизації на ряду з часом до автоматизації. Цей розділ показав нам наскільки це важливо для сучасного бізнесу. На основі цих даних я вважаю, що розвиток автоматизації може автоматизувати деякі сфери людської діяльності повністю. Я бачу перспективу розвитку у будь-якій сфері, але мета цього дослідження саме ресторанний

бізнес, у якому можлива часткова ліквідація певних посад, як бухгалтер, наприклад. Таким чином, розвиток автоматизації – великий крок вперед, який треба тільки удосконалювати.

Розділ II. Вибір типу програмної платформи.

I. Вступ до розділу.

Framework PHP – це програмні платформи, які значно полегшують та прискорюють розробку сайтів, web - та мобільних додатків. Вони частково автоматизують процес написання коду, спрощують формування структури майбутнього програмного рішення та спрощують об'єднання програмних компонентів проекту. Також у багатьох із них передбачена можливість швидкої інтеграції та налаштування взаємодії програмного продукту зі сторонніми сервісами.

Завдяки особливостям структури PHP розробникам доступна концепція Rapid Application Development – швидкої розробки додатків. Основний упор тут зроблено на швидкість роботи та зручне середовище для програмування, при цьому фахівець не обмежений у можливостях.

Одна з важливих переваг PHP-фреймворків - невисокий поріг входження для фахівців-початківців, якщо порівнювати з іншими мовами програмування. Це дозволяє навіть недосвідченому фахівцю швидко освоїти основний функціонал та оптимізувати власний робочий процес за їх допомогою.

II. Переваги та недоліки frameworks.

1). CodeIgniter.

Codeigniter - це фреймворк PHP за допомогою архітектури Model View Controller (MVC). З точки зору початківця, це означає, що Codeigniter використовує різні компоненти для вирішення конкретних завдань розвитку. Цей підхід популярний серед розробників, оскільки він дозволяє створювати масштабовані веб-програми з меншими розмірами.

Codeigniter може бути чудовим вибором, адже це досить просто вивчити та почати роботу. Крім того, платформа може похвалитися чудовою документацією, а це означає, що у вас не слід мати особливих проблем із вивченням основи. Codeigniter також забезпечує високу продуктивність, а це означає, що він ідеально, якщо ви хочете створити світлові програми, які добре працюють на скромних серверах.

Ключові особливості:

- Використовуйте створені легкі рамки з урахуванням продуктивності
- Почніть швидко, завдяки простоті рамок та відмінній документації
- Створіть масштабовані програми за допомогою архітектури MVC на основі інфраструктури.
- Існує багато переваг у використанні Codeigniter. Тим не менш, не можна назвати його найкращим фреймворком PHP, адже він також має деякі недоліки. Наприклад, його оновлення дещо нерегулярні, тому це не може бути найкращим варіантом для додатків, які потребують високого рівня безпеки.

Переваги:

- Відмінна документація та англійська, -наявна спільнота.
- Високопродуктивні фреймворки.
- Невеликий розмір фреймворку.
- Забезпечує прості та прості рішення щодо розробки.
- Підходить для швидкого розвитку невеликих сайтів та веб -додатків.
- Структура фреймворку не вимагає суворого кодування.
- Це не вимагає складних налаштувань, майже нульову конфігурацію.
- Веб -додатки Architecture MVC.
- Багато пов'язаних бібліотек та помічників.

Недоліки:

- Довгий застій у розробці Codeigniter 3.

2). Symfony.

Symfony дуже гнучкий. Він включає систему пакету та компонентів, яка дозволяє вибрати необхідні функції PHP або просто використовувати всю інфраструктуру.

Для того, щоб ваші програми працювали належним чином, платформа також включає в себе побудовану функцію тестування. Однак, програми, створені за допомогою Symfony, часто не мають найкращих показників. Плюс до цього, в цій структурі досить важко з'ясувати порівняно з іншими варіантами.

Ключові особливості:

- Використовує дуже гнучку структуру, яка дозволяє налаштувати окремі компоненти,
- Використовує переваги вбудованої функції тестування платформи,
- Symfony може бути хорошим варіантом, якщо ви піклуєтесь про модальність. Ви можете просто використовувати потрібні компоненти, замість повної бібліотеки, яка може зробити це найкращим середовищем PHP, якщо вам потрібно лише мало допомоги.

За допомогою Symfony було розроблено багато великих проектів:

- Системи управління вмістом: Magento, Drupal, OpenCart,
- Служба соціальних закладок Delicious,
- Французьке відео хостинг Dailymotion,
- Двигун форуму PHPBB.

Переваги:

- Потужна екосистема навколо рамок, з хорошою спільнотою та багатьма розробниками.
- Хороша та постійно оновлена документація для всіх версій.
- Багато різних непов'язаних компонентів для повторного використання.
- Він пропонує функціональний та модульний механізм тесту для пошуку помилок у веб -додатку.
- Підходить для складних та завантажених веб -проектів. Електронна комерція.

Недоліки:

- Незважаючи на хорошу документацію, фреймворк важко вивчити.

3). Zend.

Багато людей називають Zend "з'єднанням" фреймворком, що є способом посилання на його компонентну природу. Zend-це об'єктно-орієнтована на основі інфраструктури MVC, яка дозволяє завантажувати лише ті компоненти, які ви хочете, як окремі бібліотеки.

Іншими словами, Zend дозволяє зосередитись лише на компонентах та функціях, які бажаєш, і ігнорувати все інше. Завдяки такому підходу та об'єктно-орієнтованому характеру фреймворку ви можете повторно використовувати великий обсяг коду, який ви написали, що завжди є хорошою новиною. Крім того, інтегрувати платформу із зовнішніми бібліотеками досить просто, щоб ще більше розширити її функціональність.

Ключові особливості:

- Використовує об'єктно-орієнтоване середовище PHP з архітектурою MVC,
- Повторно використовує свій код завдяки дизайну платформи,
- Легко інтегрувати Zend із зовнішніми бібліотеками,
- Використовуйте лише ті компоненти, які ви хочете,
- Однак у Зенда є кілька недоліків. Наприклад, оволодіти платформою досить важко. Якщо ви переходите від такого інструменту, як Laravel до Zend, перехід може бути досить складним, і це може бути дещо страхітливим, якщо це ваша перша структура. Тим не менш, Zend може бути найкращим середовищем PHP, якщо є досвід розвитку, і вам подобається об'єктно-орієнтоване кодування.

Переваги:

- Відмінно підходить для розробки комерційних веб -додатків.
- Об'єктно-орієнтований підхід до розвитку.
- Непов'язані компоненти для повторного використання в проектах.

Недоліки:

- Не підходить для швидкого розвитку проекту.

- Для сегмента розробників української мови розробників мало корисних матеріалів для розвитку.

4). Phalcon.

Phalcon трохи дивний, коли справа доходить до PHP framework. Його вихідний код написаний на C, тому це в основному розширення з C в PHP. Це звучить дивно, але на практиці це призводить до одного з найшвидших середовищ.

Коли доходить до продуктивності, Phalcon виправдовує своє ім'я, і він послідовно забезпечує результати, які знаходяться на вершині. Phalcon також дуже легкий у використанні ресурсів та використовує архітектуру MVC. Крім того, він унікальний тим, що сам фреймворк практично не має файлів після встановлення. Натомість ви просто додаєте модулі та бібліотеки, які вам потрібні, коли вони вам потрібні, що робить процес розробки дуже простим.

Ключові особливості:

Використовуйте PHP-фреймворк на основі C

- Скористайтеся перевагами фантастичної продуктивності Phalcon та низькими витратами ресурсів,
- Використовуйте лише ті модулі та бібліотеки, які вам потрібні,
- Поки що Phalcon виглядає досить добре. Але варто відзначити, що, оскільки ми говоримо про менш популярний інструмент, його документація не відповідає стандартам таких гігантів, як Laravel. Пошук допомоги з помилками також може вимагати більше зусиль. Тим не менш, Phalcon досить близький до того, щоб бути найкращим PHP framework, який ви можете використовувати, особливо якщо вам потрібна продуктивність.

Переваги:

- Висока продуктивність,
- Невелике завантаження файлової системи,
- Менше споживання пам'яті,
- Часткова обробка даних без інтерпретації.

Обмеження:

- Якщо ви не знаєте с, то ви повністю залежні від розробників,
- Проект можна підірвати на будь -яку секунду,
- Залежно від архітектури, деякі модулі все ще повинні писати собі, що зменшує вираш.

5). CakePHP.

На початку 2000 -х CakePhp був першим кадром PHP MVC, який з'явився на ринку. Тоді це було відкриття, і це все ще одна з найкращих платформ PHP, яку можна використовувати (і одна з найпопулярніших).

Нові версії CakePhp з часом покращилися та додали багато нових компонентів. Однак там, де CakePhp дійсно світить, - це підхід до кодування. Це означає, що з CakePhp, як тільки ви освоїте його набір угод, ви можете зосередитись на розробці та швидше виконувати роботу.

Поряд з цим, CakePHP пропонує чудовий набір бібліотек, який включає багато корисних компонентів (як це має бути в будь -якому середовищі PHP). Під час наших тестів ми виявили, що CakePhp пропонує один із найповніших наборів компонентів, які ви можете знайти, тому це чудовий варіант, якщо вам потрібно реалізувати функції жорстких до -реєстрації.

Ключові особливості:

- Використовуйте переваги широкого набору компонентів,
- Використовуйте угоди CakePHP для швидшого кодування проектів,
- Звичайно, CakePhp може бути трохи обмеженим через його любов до угод. Якщо ви любите кодувати та показувати найкреативніший підхід, CakePhp може бути не найкращим варіантом для вас. Однак це хороший вибір для багатьох розробників.

Переваги:

- MVC: CakePHP забезпечує моделі; Конкретний клас моделі, яка може бути використана для вставки, оновлення, видалення або читання даних з бази даних. Підсистема машинного зору обробляє передачу даних на екрані, а контролер обробляє події та реагує на нього, а також

обмінюється даними перед взаємодією з моделями. Така система дозволяє легко відокремити логіку веб -програми від презентації, яка прискорює розробку великих додатків та складних веб -сайтів.

- Об'єктно-революційне порівняння: CakePHP підтримує об'єктивне реляційне порівняння, яке спрощує створення веб-додатків для розробників.
- Автентична автентифікація: задоволення від розробки CakePHP лежить у мінімальних зусиллях, витрачених розробником для налаштування системи.
- Розширення: Однією з найбільших переваг сучасних мов програмування є те, що їх можна розширити за допомогою модулів, бібліотек та плагінів. Це гарантує, що для більшості коду розробнику не потрібно "вигадувати велосипед" та писати функції та класи з нуля. Модульний характер CakePHP дозволяє розробникам повторно використовувати код, надбудови та компоненти, які легко інтегруються в платформу за допомогою набору інструментів автоматичного виявлення.
- Підтримка Ajax. Сучасні веб -програми часто потребують обміну даними між клієнтом та сервером, навіть якщо програма працює в браузері. Традиційне PHP оновлює сторінку щоразу, коли нові дані надходять з сервера: величезний недолік для розробки веб -додатків, таких як електричні таблиці, текстові процесори та поштові клієнти. AJAX забезпечує асинхронний обмін даними і став стандартною практикою для веб -розробників при створенні багатофункціональних та захоплюючих веб -додатків. CakePhp принципово підтримується AJAX і може використовуватися різними способами, включаючи форми, події та навіть модельні події відносин.

б). Yii.

Yii - ще один варіант нашого огляду найкращих рамок PHP, і він є гідним конкурентом. Одразу ми можемо сказати вам, що Yii світить через те, як легко

його налаштувати. Поряд з цим, рамки знаходяться на одному рівні з Phalcon, коли мова йде про продуктивність, що є величезною перевагою для продажу. Як і інші рамки PHP, які ми бачили досі, Yii пропонує надійний набір компонентів, які можна використовувати для прискорення розробки веб - додатків. Тим не менш, він виділяється завдяки потужному набору функцій безпеки, які можна використовувати для створення проектів з високим ступенем захисту.

Ключові особливості:

- Чудова швидкість та продуктивність,
- Використовуйте налаштування за замовчуванням або налаштуйте їх відповідно до ваших потреб,
- Створіть безпечні сайти та програми,
- Що стосується недоліків, то Yii може бути трохи складним, якщо це ваша перша рамка. Це головний конкурент за назвою найкращої рамки PHP, але якщо ви погано знайомі з мовою кодування, є й інші, зручніші для початківців, якими ви можете використовувати.

Переваги:

- Для російських розробників, великий плюс рамки -це хороша документація, багато статей із прикладами коду та спільнотою.
- Yii не є показником однієї людини, фреймворк підкріплюються сильною командою основних розробників, а також великою спільнотою професіоналів, що постійно сприяють розвитку Yii.
- Yii сприяє швидкому прототипуванню веб -додатку. Він належить до інструментів розвитку рейду.
- Побудований механізм створення віджетів представництва, наприклад, для розміщення різних блоків на сайті: останні публікації, категорії, навігація, рекламні блоки тощо.
- Компонент програми i18n дозволяє автоматично перекладати веб - програми.

- Вбудована підтримка автоматичної перевірки форм та виведення повідомлень про помилки на основі даних з веб-додатків.
- Активний механізм запису побудови реляційної обробки запитів на базу даних.
- Багато готових розширень на Github та їх встановлення за допомогою композитора
- Побудовані -віджети для відображення даних: DetailView (рядки в таблиці), ListView (LIST), GridView (таблиці)
- Побудовані механізми аутентифікації, авторизації, реєстрації користувачів
- Містить побудовану та дуже зручну панель налагодження.

Недоліки:

- Слабка екосистема навколо фреймворку серед англійської мови сегмента розробників. Громада занадто змащена в різних місцях: кілька форумів з незначною активністю, Stackoverflow тощо.
- Хоча фреймворк дозволяє зробити код простим, але далеко не елегантним. Якщо його синтаксис порівнюється з рамкою Laravel, то він неповноцінний.
- Yii стоїть за мовою, стандартами та іншими рамками. Нові оновлення з дійсно корисними функціями не виходять так часто.
- Занадто багато контакту бекенду та фронтенду частин Yii2. Freimvork пропонує використовувати бібліотеку jQuery та Bootstrap, яка вбудована в ядро Framvorka за замовчуванням. Цей мінус планується виправити в новій версії рамки YII 3, що робить його компоненти менш пов'язаними.
- Не дуже гнучка система маршрутизації: немає можливості групувати маршрути.

7). FuelPHP.

FuelPHP - це відносно нова рамка PHP, вперше запущена в 2014 році. Творці стверджують, що вони зібрали найкращі практики з інших потужних рамок і

створили щось нове та захоплююче. Деякі з найбільш унікальних аспектів описані у функціях функцій (англійська).

Ключові особливості:

- Повна підтримка HMVC,
- Покращена безпека за замовчуванням,
- Унікальна утиліта командного рядка,
- Рамка підтримує PHP версію 5.4 і вище. Крім того, він пропонує детальну документацію (англійською мовою), яка допоможе почати розвиватися якомога швидше.

8). Laravel.

Laravel - це безкоштовний фреймворк PHP з відкритим кодом, створеним Taylor Wheller для розробки веб -додатків відповідно до архітектурного шаблону MVC.

Він був створений як альтернатива фреймворку кодового сигналу (про неї буде описано нижче), в якій були недостатньо корисні функції для розробки веб -додатків. Компоненти чергової рамки - Symfony (про це також буде описано нижче) є основою Laravel. Laravel Framework дуже популярний серед західних веб -додатків. Використовуючи менеджер пакетів композитора, Laravel дозволяє легко встановлювати та підключати різні компоненти для використання у веб -додатку. Впровадження шаблону ActiveRecord - красномовний ORM дозволяє встановити взаємозв'язок між об'єктами бази даних веб -додатків та створити зручні запити для маніпулювання даними.

Зручна міграційна система допомагає спростити розгортання та оновлення веб -програми. Laravel має вбудовану підтримку шаблонів, за допомогою яких ви можете зробити прості перегляди веб-програми за допомогою спеціального синтаксису. Створюючи додаток, ви можете використовувати інтерфейс Artisan командного рядка для введення вбудованих команд, а також створення власного. Laravel має багато корисних функцій, щоб зробити процес розробки веб-додатків швидкими, простими та якісними. З цих функцій можна відзначити `dd()` - зручний аналог стандартної функції `php var_dump()`. Функція відображає

змінну інформацію в більш зрозумілому вигляді, ділячи дані на дерево атрибутів та значень, у можливості їх пошуку та перетинання.

Переваги:

- Досить хороша і зрозуміла документація.
- Навколо фреймворку створена потужна екосистема. Різні курси, конференції, навчальні матеріали дозволяють зібрати велику кількість розробників та спонсорів навколо, які зацікавлені в розробці інструменту та участі в цьому.
- Однією з найбільш очевидних переваг Laravel є гнучка система маршрутизації, яка дозволяє скласти різноманітні маршрути веб - додатків. Ви можете виділити маршрути на спеціальні групи, використовувати простір імен, вказати параметри маршруту, використовувати регулярні вирази, налаштувати підлеглу маршрутизацію та багато іншого.
- У Laravel синтаксис API рамки досить простий і зрозумілий. Немає довгих і складних конструкцій, але лише коротких і продуманих назв функцій.
- Laravel містить зручний механізм для обробки помилок та винятків.
- Laravel включає вбудовані механізми аутентифікації та авторизації користувачів, які можна переналаштувати для їх потреб.
- Laravel надає механізми коробки для кешування веб - додатку за допомогою Memcache та Redis. Крім того, існують зручні функції для використання простих кешування даних файлів.
- Laravel надає чистий і простий API поверх популярної бібліотеки SwiftMailer з драйверами для SMTP, Mailgun, SparkPost, Amazon SES та Sendmail, щоб надіслати пошту через локальну службу вибору хмар. У тому числі є механізм побудови черг надсилання пошти.
- Laravel Cashier надає виразний, безкоштовний інтерфейс для виставлення рахунків Stripe та Brinetre.

Недоліки:

- Для російських розробників, без знання англійської мови чи її слабких знань, досить невелику кількість статей, прикладів коду та перекладу офіційної документації можна віднести до мінусів. Для тих, хто знає англійську на рівні читання технічної документації, цей мінус можна опустити.
- Синтаксичний цукор у Laravel як плюс, тому це може бути мінус. Накручити до нього дуже легко і забути, як пишуться чисті запити та функції.
- Порушення зворотної сумісності між версіями рамки.
- Не логічне розташування каталогів та файлів. Наприклад, за замовчуванням модель `USR.PHP` розташована безпосередньо в каталозі `/додатки`, який був би більш логічним, щоб розташовуватися в каталозі `/app /Models`. Каталог ресурсів із файлами презентації знаходиться в корені програми, хоча буде більш логічно розміщувати його в `/додатки /ресурси`.

III. Порівняння фреймворків.

	Laravel	Yii	Symfony	Zend	Codeigniter
Ліцензія	MIT	BSD	MIT	New BSD License	MIT
Документація	добра	відмінна	добра	непогана	найкраща
Спільнота	Сильна англійська, українська слабка	Сильна російська, англійська слабка	Сильна англійська	Сильна англійська	Сильна англійська
Доп. Пакети та розширення	+	+	+	+	+
Установка за допомогою Composer	+	+	+	+	+
PHP 7	+	+	+	+	+

Бази даних	MySQL, Postgres, SQLite и SQL Server	MySQL, MariaDB, SQLite, PostgreSQL, CUBRID, Oracle, MSSQL	MySQL, PostgreSQL, SQLite или любая другая PDO-совместимая СУБД)	MariaDB, MySQL, Oracle Database, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite и Informix	MySQL, PostgreSQL, MSSQL, SQLite, Oracle.
Міграції	+	+	+	+	+
Захист від SQL- injection, XSS, CSRF	+	+	+	+	+

IV. Висновок з розділу.

- В даний час Laravel є найбільш перспективним фреймворком PHP і підходить для створення як малих, так і великих веб -проектів.
- Ви все ще можете брати участь у розробці нових проектів на Yii2, а також продовжувати вести існуючі проекти. Версія 3 все ще розробляється, тому Yii знаходиться в проміжку, і краще взяти для цього ще одну рамку для цього, наприклад, Laravel.
- Symfony та Zend найкраще використовуються для великих комерційних проектів з тривалим періодом розвитку.
- Дуже важливим фактором розвитку фреймворків є комерційний посібник навколо них. Якщо у нього немає спонсорів, то рано чи пізно команда розробників просто втратить інтерес до участі у розвитку рамок. Значна роль у розвитку фреймворку - це маркетинг. Немає популярності та піару навколо фреймворку, а це означає, що немає розробників, зацікавлених у розвитку фреймворка.
- PHP - це незамінний інструмент для швидкої зведення фундаменту будь - якого сайту. Вони допоможуть прискорити процес розробки та зробити якісний проект, уникаючи програмування всіх частин з нуля.
Використовуйте або не використовуйте - це ваше рішення.

IV. Розділ III. Опис основних модулів.

I. Модульне програмування та структура:

Модульна структура архітектури проекту є чудовим рішенням для великих та розроблених додатків. Але, якщо ми поговоримо про Laravel, то немає коду, який би організував такий спосіб з "коробки". Тому зараз ми порушимо питання: «як додати підтримку модальності до програми, написаної за допомогою Laravel.»

Я хотів би зазначити, що модульне програмування - це організація коду проекту, як поєднання багатьох окремих елементів модуля. Більше того, як правило, кожен з них має свою функціональність та можливості.

За замовчуванням в Laravel, зазначена можливість не закладена, і тому вам доведеться створити її вручну. Звичайно, в цей час вже є готові пакети, які додають функціональність, але я хочу показати вам, як його реалізувати самостійно.

Особливості модульного програмування:

- Модульні програми легко складати та налагодити.
- Функціональні компоненти такої програми можна записати та налагодити.
- Модульну програму простіше супроводжувати та змінювати.
- Функціональні компоненти можна змінити, переписати або замінити без змін в інших частинах.
- Керівництву простіше керувати розробкою модульної програми. Більш складні модулі можуть бути передані більш досвідченим програмістам; Прості модулі можуть написати молодшими програмістами. Розбиваючи програму на модулі, які можна створити за один місяць, лідер може бути впевнений, що жоден із програмістів не буде занадто перевантажений складними елементами програми.
- Модульну програму простіше налагодити, оскільки модулі можуть піддаватися "автономному" тестуванню та налагодженню, тобто. Кожен

модуль може пройти тестування та налагодження окремо на підготовленому наборі тестових даних;

- Але найголовніше - це використання бібліотек «готових» модулів, орієнтованих на вирішення певного класу наукових та технічних проблем.

Переваги, які витікають з особистостей:

- Прискорення розвитку.
- Підвищення надійності.
- Спрощення тестування.
- Взаємозамінність.

Недоліки:

- Модульність вимагає додаткової роботи. Для написання модульних програм програміст повинен бути набагато акуратнішим на етапі розробки програмного забезпечення. Він повинен розробити свої програми відповідно до схеми спуску, починаючи з верхніх рівнів усієї програми, а потім рухаючись до більш детальної конструкції окремих підпрограм.
- Модульний підхід іноді вимагає більше часу. Ця проблема виникає в першу чергу у випадках, коли програма відрізняється наявністю великої кількості підпрограм, написаних мовами високого рівня. Для забезпечення модульності може також зайняти більше часу процесора, якщо частини програми з командами введення та виводу повністю відокремлені від його обчислювальних частин; Запис введення може передаватися кілька разів до підпрограм, перш ніж розпочнеться його фактична обробка.
- Модульний підхід може забрати трохи більшої кількості пам'яті. Якщо кожній підпрограмі надається окрема частина робочої пам'яті, то вся програма може забрати трохи більшої кількості пам'яті. У випадках, коли програма реалізується на основі великої кількості підпрограм, може знадобитися додаткова пам'ять для забезпечення з'єднань між ними.

II. Аналіз пакетів.

Можна використовувати готові пакети для створення модульної структури, деякі з них ми зараз проаналізуємо. Також ці пакети надають певні функції.

1). Active eCommerce CMS.

Цей популярний CMS має все, що було створено для створення сайту електронної комерції ASU до Maintain. З тисячами продажів, високими відгуками клієнтів та регулярними оновленнями, щоб скористатися останніми функціями Laravel.

2). Acelle.

Веб-програма для маркетингу електронної пошти, написана в Laravel, Aceelle дозволяє користувачам надсилати електронні листи з великим об'ємним через власні постачальники серверів. Це означає, що користувачі мають повний контроль над усією системою маркетингу електронної пошти та можуть замінити дорогий маркетинг електронної пошти, як MailChimp.

Ключові особливості:

- Керуйте списком та передплатниками,
- Інтеграція з власним сервером SMTP,
- Повністю представлена автоматизація та автовідповідач,
- і більше...

3). Josh.

Джош-це шаблон адміністратора Laravel, який має компонент, необхідний для створення додатків Back-end. Шаблон оснащений більш ніж 100 сторінками, але якщо вам потрібно лише кілька сторінок для вашої програми, шаблон включає зручний комплект для стартера, який включає найнеобхідніші сторінки, такі як вхід, реєстрація, забутий пароль та управління користувачами. Якщо вам потрібні додаткові сторінки Besse, надані в наборі стартера, ви можете отримати їм основну версію Джоша.

Ключові особливості:

- Сторінки задніх та передових сторін з головними макетами,
- Вхід та реєстрація функціоналістики,

- Модуль блогу з коментарями,
- і більше...

4). MTDb: Ultimate Movie and TV Database.

MTDB-це багатоцільовий сценарій PHP, заснований на рамках Laravel, яку ви можете швидко створити веб-сайт фільму чи телесеріалу. Сайт може запропонувати різні види функцій, виходячи з того, що ви маєте намір зробити з цим.

Ключові особливості:

- Проста установка,
- Настроювана домашня сторінка,
- Управління контентом,
- Темний режим,
- Платіжні шлюзи,
- та багато інших функцій...

Ви можете використовувати його для створення веб -сайту, подібного до IMDb або потокової служби, як Netflix. Ви також можете створити веб -сайт агрегатора, який посилається на безкоштовні потокові посилання по всьому Інтернету.

5). LaraClassifier: Classified Ads Web Application.

Laraclassifier-це потужний, повністю чуйний веб-додаток для класифікованих оголошень з відкритим кодом. Laraclassifier є SEO-оптимізованим, і ви не можете створювати списки каталогів, класифікації нерухомості, класифіковані роботи тощо.

Ключові особливості:

- Подання переднього кінця,
- Підтримка гео-локації,
- Логін соціальної мережі,
- Захист для користувачів,
- Підтримка формату Image Webp.

Він підтримує кілька мов, включаючи мови, орієнтовані на RTL. Інтегрується з Google AdSense, Google Maps, Recaptcha, коментарями у Facebook та API для текстових повідомлень, таких як Twilio та Nexmo.

6). BeDrive.

Якщо ви хочете створити власну версію Box або Google Drive, перевірити BeDrive. Цей пакет Laravel дозволяє створювати власний веб-сайт для обміну та розміщення файлів за лічені хвилини, не маючи знань кодування.

Ключові особливості:

- Інтуїтивна інформаційна панель,
- Налаштована та проста система підписки,
- Панель адміністратора,
- Google Analytics інтегрована,
- і більше...

7). Buzzy.

Buzzy-це ще один варіант пакету Laravel для того, щоб ви створили власний медіа-обмін. Він пропонує користувачам необмежену кількість віджетів, сторінок, категорій та можливості створювати представлені публікації, списки, опитування, відео тощо. Усі засоби масової інформації можуть пригнічувати щодня, щотижня чи щомісяця, щоб адміністратор сайту знала, що є найпопулярнішим та цікавим для відвідувачів.

Ключові особливості:

- Налаштовані макети,
- Панель адміністрації заднього кінця,
- Інтеграція з різними каналами соціальних медіа,
- і більше...

8). FleetCart: Laravel eCommerce System.

FleetCart-це система електронної комерції SEO-оптимізованої SEO-оптимізованої, яка дозволяє легко розпочати бізнес. З його потужної панелі адміністратора ви каналізуєте продукти, категорії, купони, замовлення,

сторінки, доставку тощо. Система захищена від ін'єкцій SQL, XSS, CSRF та інших атак.

Основні особливості включають:

- Завантажувач файлу перетягування,
- Імпорт продукції,
- Розширений пошук та просування фільтрації продуктів,
- Варіанти та атрибути продукту,
- Кілька місцевостей, країн та валют,
- Панель клієнтів та каси гостей,
- Підписка.

9). Geo POS.

Geo POS - це повне рішення з продажу для людей, які беруть на себе дещо, що забезпечує підтримку як для управління відносинами з клієнтами (CRM), так і для управління людськими ресурсами (HRM). Він доступний на 21 різних мовах і підтримує кілька шлюзів оплати.

Ключові особливості:

- Підписки,
- Виставлення рахунків,
- Управління запасами,
- Управління клієнтами,
- Система підтримки на основі квитків.

10). Worksuite: HR, CRM, and Project Management.

Workuite автоматизує ділові завдання, щоб ви могли зосередитись на веденні свого бізнесу.

Основні особливості включають:

- Управління клієнтами та лідерами,
- Отримання платежів через PayPal, Stripe та Razorpay,
- Створіть трек та керуйте проектами,
- Внутрішні командні повідомлення,
- Управління квитками на підтримку клієнтів.

11). Vanguard

Vanguard - це програма для управління входу та управління користувачами, яка дозволяє власникам веб -сайтів швидко додавати та включати функції аутентифікації та авторизації користувачів на свій веб -сайт. Він забезпечує просту інтерактивну панель приладів, потужну панель адміністратора та необмежену кількість ролей користувачів. Він також постачається з повністю задокументованим API JSON, який дозволяє легко аутентифікуватись за допомогою улюбленого мобільного пристрою.

Ключові особливості:

- безпечна реєстрація користувачів та вхід,
- Соціальна автентифікація за допомогою Facebook, Twitter та Google,
- Скидання пароля,
- Google Recaptcha щодо реєстрації,
- і більше...

12). Booking Core.

Booking Core - це чудовий вибір для людей, які шукають дещо, що забезпечує всі необхідні функції для створення повністю функціонального веб -сайту для подорожей та для вашого бізнесу. Ви можете нам надати такі послуги, як ринок, оренда курортів, туристичне агентство тощо.

Ключові особливості:

- Приладна панель адміністратора,
- Builder сторінки та меню,
- Управління відгуками,
- Оплата,
- Вбудована локалізація,
- і більше...

13). eClass Learning Management System.

Цей Eclass LMS - це прекрасний спосіб почати створювати та продавати курси в Інтернеті. Він поставляється з безліччю фантастичних особливостей. Ця

система управління навчанням оптимізована як формування, так і для пошукових систем.

Ключові особливості:

- Зручний для клієнтів,
- Вишуканий дизайн,
- Адміністратор, інструктор та інформаційна панель студентів,
- Інтеграція масштабу,
- Платіжні шлюзи,
- і більше...

14). Rawal Laravel eCommerce.

Цей пакет електронної комерції Laravel переповнений функціями, які дозволяють продавати всі види продуктів. Ви зможете створити повністю функціональні веб-сайти з темами та передовою панеллю CMS. Пакет був розроблений для того, щоб бути максимально дружнім SEO. Також є кілька домашніх екранів.

Ключові особливості:

- Підтримка декількох мов
- Підтримка декількох валют
- Кілька ролей адміністратора
- Кілька шлюзів оплати
- Натискайте сповіщення одним сигналом
- і більше

15). Classic Invoicer.

Веб-система рахунків-фактур та управління клієнтами, Classic Invoicer дозволяє користувачам створювати власні рахунки-фактури та надсилати їх безпосередньо в клієнтам. Це також деталізує виробництво, які мають повноцінно в управлінні та відстеженні доходу.

Ключові особливості:

- Вишуканий дизайн,
- Зберігає інформацію про клієнта для кожного рахунку-фактури,

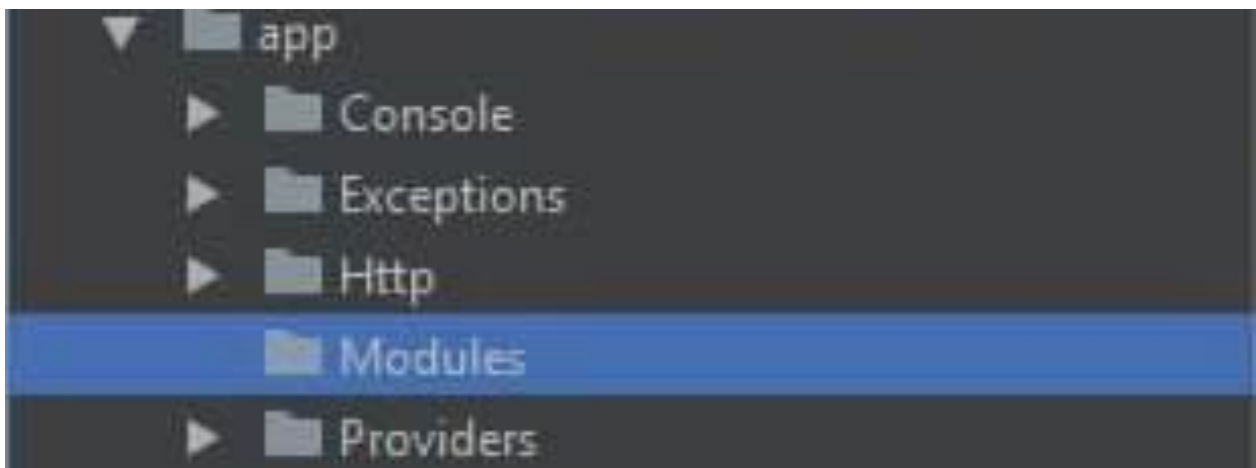
- Обробляє оцінки за допомогою програми,
- Керує різними типами платежів,
- і більше...

Таким чином ми виявили, що існує велика кількість пакетів, які тим або іншим чином впроваджують різноманітні функції до вашого проекту, що полегшує розробку та проектування.

III. Створення примітивного коду для аналізу принципів модульного проектування.

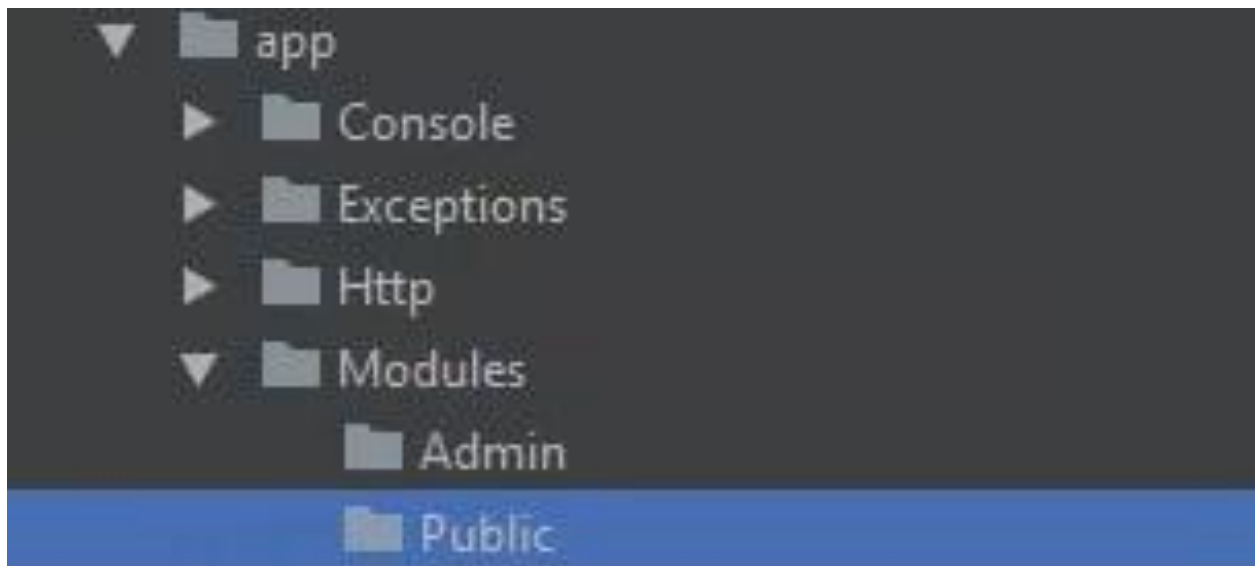
Зараз ми додамо код, який буде підтримувати роботу з окремими модулями у Laravel для розуміння сенсу та принципу роботи модулів.

-Створимо у директорії "app" каталог "Modules":



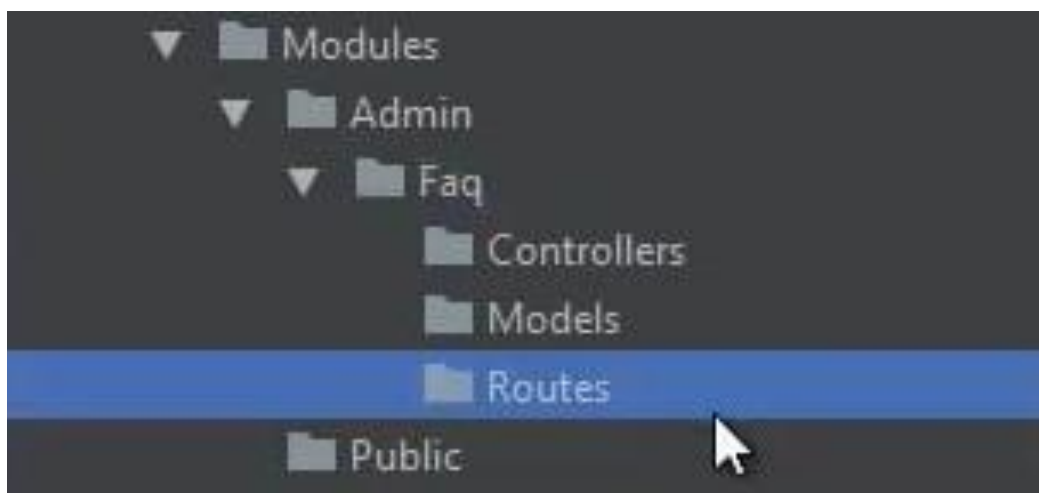
-тут будуть знаходитися модулі нашого проекту. Створимо систему модулів 2 рівней: верхні та нижні модулі. Або інакше модулі 1 та 2 рівнів. Модулі верхнього рівня не будуть виконувати якоїсь прямої функції, вони будуть лише категорією нижніх модулів, які саме і будуть виконувати певні функції.

-Створено 2 верхніх модулі "Admin" та "Public":

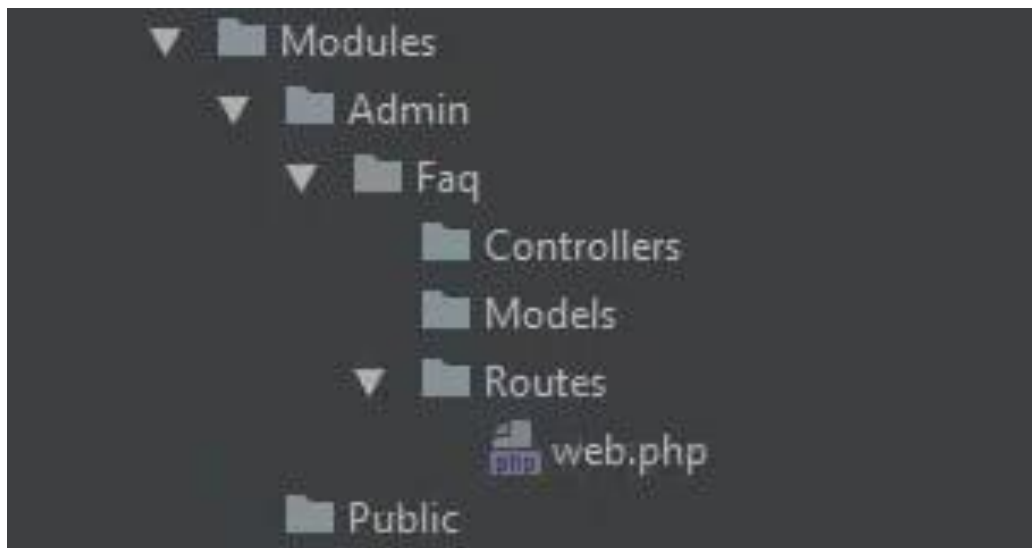


Кожен з цих модулів буде тримати у себе певну кількість дочірніх елементів, які будуть виконувати певну роботу.

-Створимо модуль "Faq", у якому створимо низку нижчих категорій, а саме "Controllers", "Models" та "Routes":



-Для директорії "Routes" створимо файл з маршрутами "web.php":



-Модель для директорії "Models" треба створювати через консоль необхідно правильно описати простір імен.

Але також не треба забувати написати саму назву моделі, щоб не було помилки, як на скріншоті:

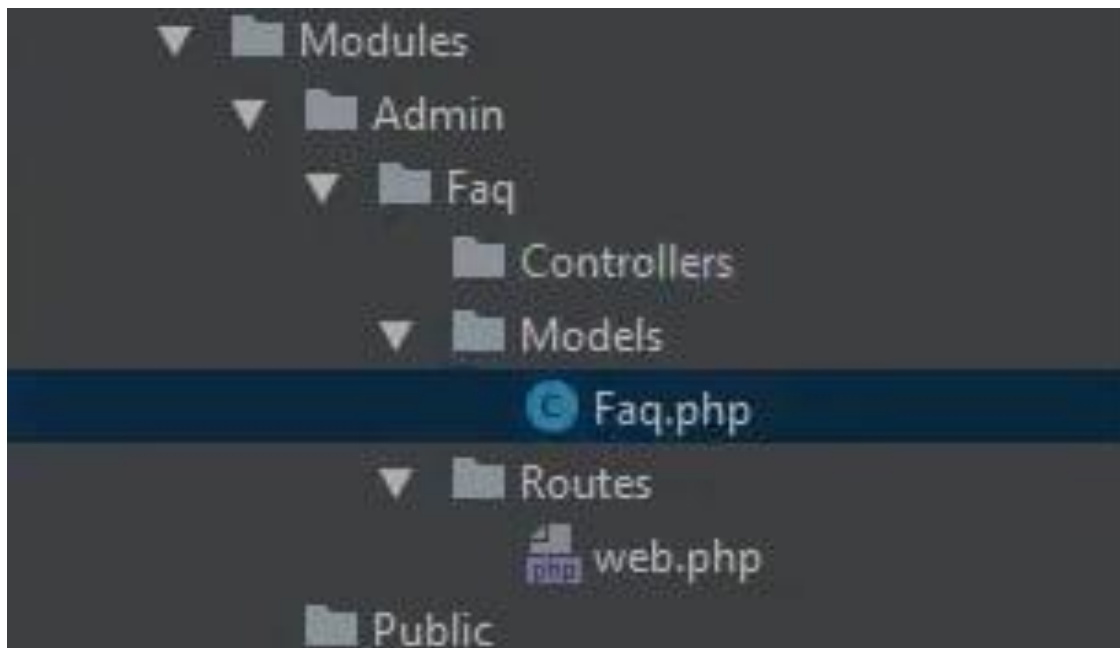
```
Microsoft Windows [Version 6.3.9600]
viktor@VIKTORNOTE d:\ospanel
> cd domains

viktor@VIKTORNOTE d:\OSPanel\domains
> cd tlaravel.loc

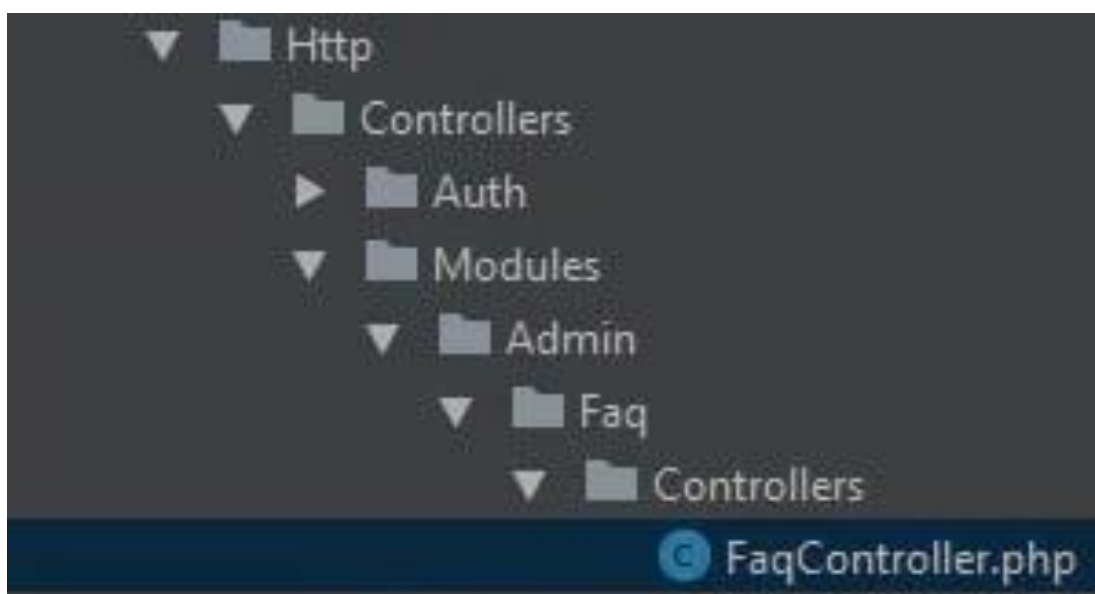
viktor@VIKTORNOTE d:\OSPanel\domains\tlaravel.loc
> php artisan make:model Modules/Admin/Faq/Models
^C
viktor@VIKTORNOTE d:\OSPanel\domains\tlaravel.loc
> php artisan make:model Modules/Admin/Faq/Models/Faq
Model created successfully.

viktor@VIKTORNOTE d:\OSPanel\domains\tlaravel.loc
> |
```

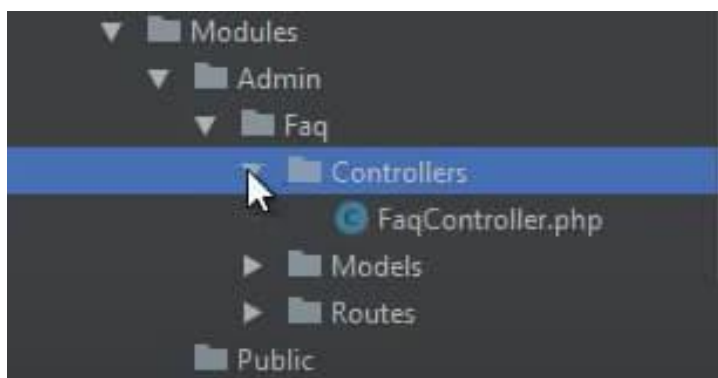
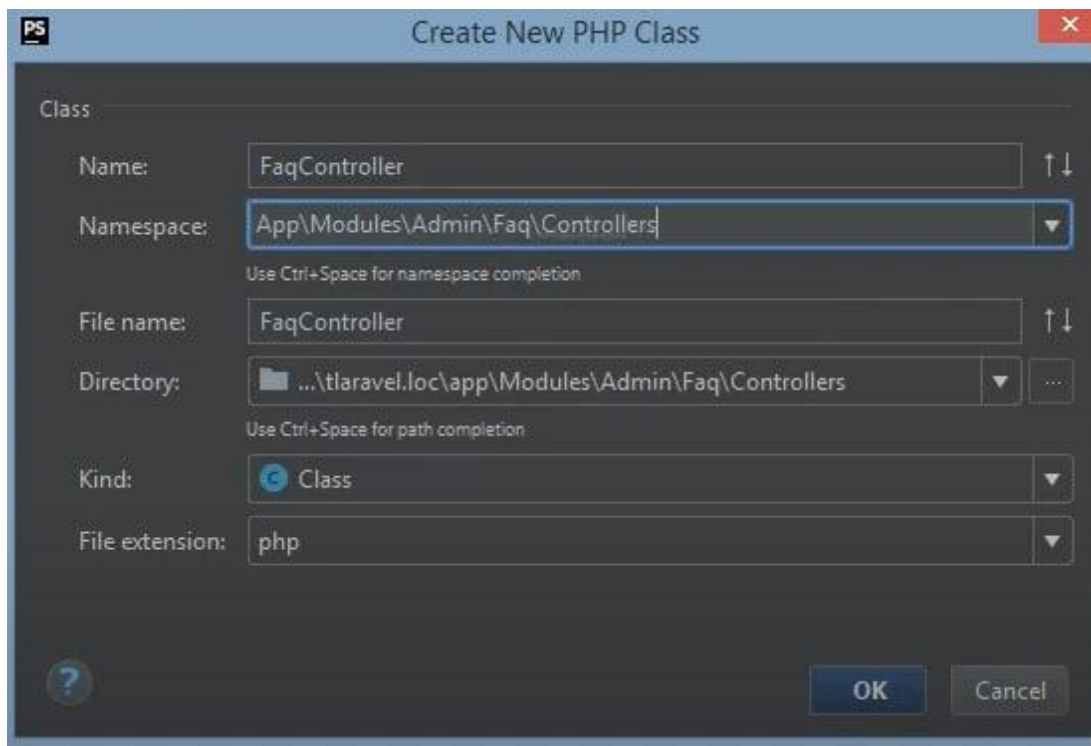
-модель створена:



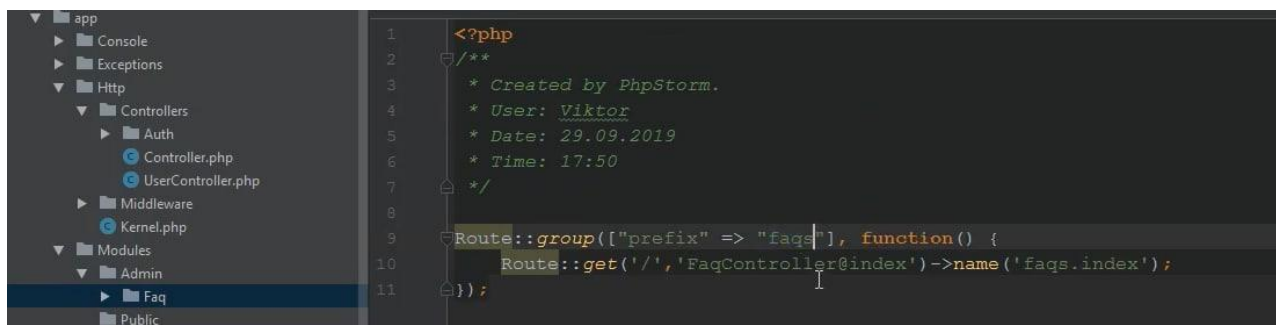
-Таким самим чином ми створюємо контроллер можна, бо він автоматично приєднається до іншої директорії:



-Тому треба створювати власноруч:



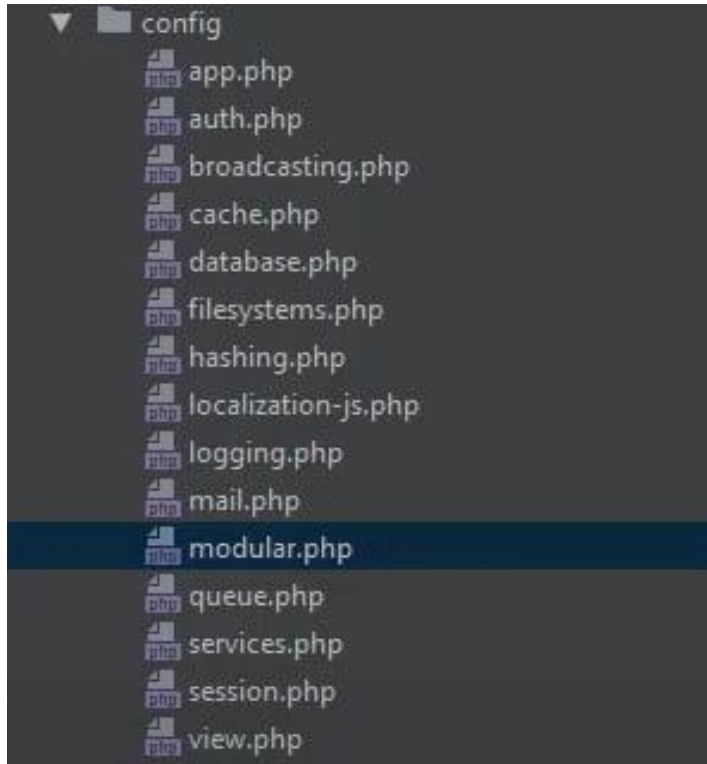
Треба задати маршрути, бо вони не визначені. Задамо prefix. 2 аргумент буде функція callback, який буде визнавати маршрути, наприклад, маршрут типу get, у якості оброблювача буде FaqController. Цей запит буде опрацьований методом index. Також можна задати псевдонім маршруту:



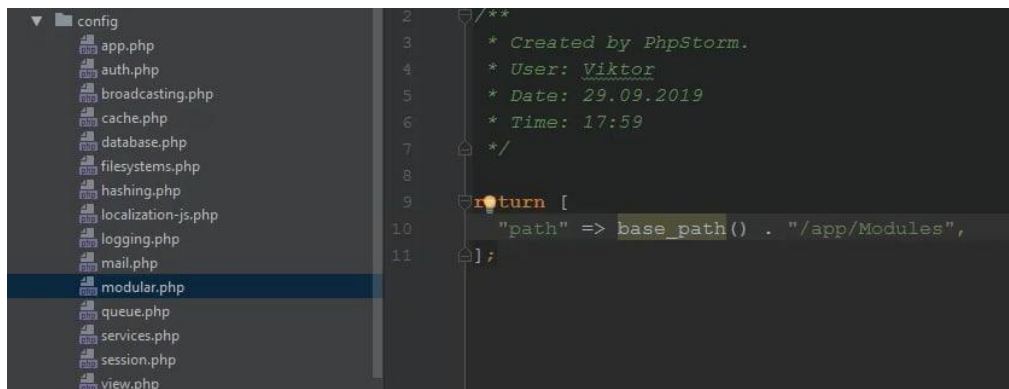
-Якщо подивитися з боку принципів роботи, наш модуль готовий, бо у нього є свої маршрути, є свій контроллер, є своя модель. Все що залишилося, ми

повинні показати фреймворку як і де звертатися до моделі, щоб запобігти виникненню помилки «404».

-У директорії "config" треба створити новий файл, наприклад "modular.php", у якому ми будемо зберігати інформацію про модулі нашого проекту:



-Через масив "path" ми повинні вказати шлях до наших модулів:



-"base_namespace" допоможе нам визначати базовий простір імен для наших модулів:

```
4  /**
3  * Created by PhpStorm.
4  * User: Viktor
5  * Date: 29.09.2019
6  * Time: 17:59
7  */
8
9  return [
10     "path" => base_path() . "/app/Modules",
11     "base_namespace" => "App\Modules",
12
13
14 ];
```

-Далі ми прописуємо вже наші безпосередньо готові модулі:

```
4  /**
3  * Created by PhpStorm.
4  * User: Viktor
5  * Date: 29.09.2019
6  * Time: 17:59
7  */
8
9  return [
10     "path" => base_path() . "/app/Modules",
11     "base_namespace" => "App\Modules",
12
13     /** Modules */
14
15 ];
```

-У модулях ми будемо зберігати modules, а вона буде зберігати у собі масив.

Тут ми вказуємо категорії модулів верхнього рівня та надаємо їм масиви:

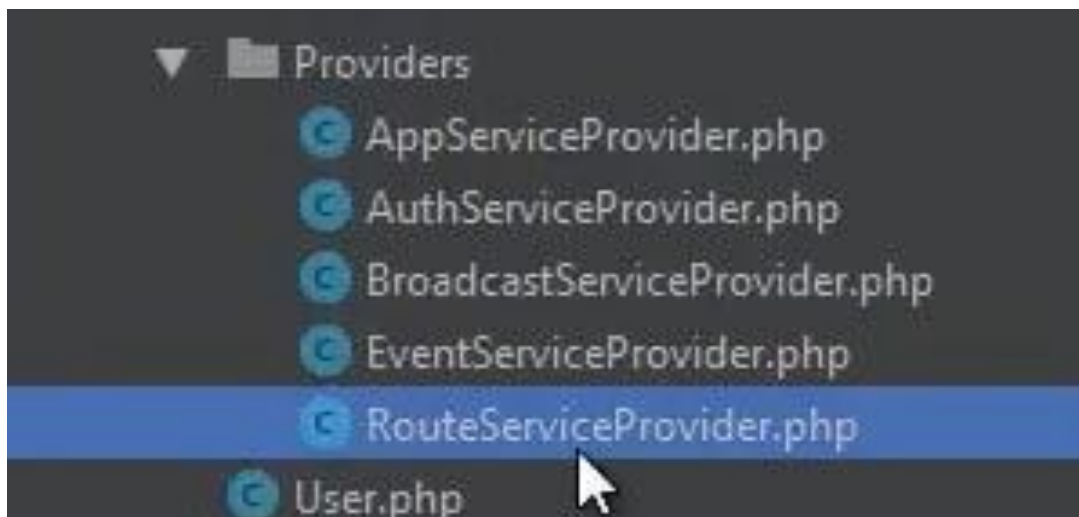
```
9  return [
10     "path" => base_path() . "/app/Modules",
11     "base_namespace" => "App\Modules",
12
13     /** Modules */
14
15     'modules' => [
16         "Admin" => [
17             |
18         ],
19         'Public' => [
20             |
21         ]
22     ]
```

-У модулі Admin позначаємо модуль Faq:

```
13  /** Modules */
14
15  'modules' => [
16      "Admin" => [
17          'Faq',
18      ],
19      'Public' => [
20
21      ]
22  ]
23  ;
```

-Переходимо до стандартного файлу маршрутів. У цьому файлі треба описати механізм пошуку та обробки модулів, тут буде описан код по пошуку маршрутів

-це можна створити у сервіс провайдер:



-Але на мій погляд більш наглядно та зручно це зробити у стандартному файлі маршрутів.

тут треба зазначити деякі змінних, які нам знадобляться, наприклад змінна modules яка буде зберігати у собі список модулів з config:

```
▼ app
  ► Console
  ► Exceptions
  ► Http
  ▼ Modules
    ► Admin
    ► Public
  ► Providers
  User.php
  ► bootstrap
  ▼ config
    app.php
```

```
1 <?php
2
3 /*...*/
13
14 Route::get('/', function () {
15     return view('welcome');
16 })->name('home');
17
18
19 $modules = config("modular.modules");
```

-знадобиться змінна path та base_namespace:

```
1 <?php
2
3 /*...*/
13
14 Route::get('/', function () {
15     return view('welcome');
16 })->name('home');
17
18
19 $modules = config("modular.modules");
20 $path = config("modular.path");
21 $base_namespace = config("modular.base_namespace");
22
23
```

-треба поставити if для виконання коду та циклу, зробимо перевірку, як заготовку до наступного удосконалення та розширення функціоналу:

```
18
19 $modules = config("modular.modules");
20 $path = config("modular.path");
21 $base_namespace = config("modular.base_namespace");
22
23 if($modules) {
24     foreach ($modules as $mod => $submodules) {
25         foreach ($submodules as $key => $sub) {
26             if(is_string($key)) {
27                 $sub = $key;
28             }
29         }
30     }
31 }
```

-зараз треба написати шлях до нашого модулю. створюємо змінну, яка буде зберігати шлях до нашого модулю:


```

19 $modules = config("modular.modules");
20 $path = config("modular.path");
21 $base_namespace = config("modular.base_namespace");
22
23 if($modules) {
24     foreach ($modules as $mod => $submodules) {
25         foreach ($submodules as $key => $sub) {
26             if(is_string($key)) {
27                 $sub = $key;
28             }
29
30             $relativePath = "/" . $mod . "/" . $sub;
31         }

```

-також треба показати де брати маршрути через змінні:

```

19 $modules = config("modular.modules");
20 $path = config("modular.path");
21 $base_namespace = config("modular.base_namespace");
22
23 if($modules) {
24     foreach ($modules as $mod => $submodules) {
25         foreach ($submodules as $key => $sub) {
26             if(is_string($key)) {
27                 $sub = $key;
28             }
29
30             $relativePath = "/" . $mod . "/" . $sub;
31             $routesPath = $path . $relativePath . "/Routes/web.php";
32         }
33     }
34 }

```

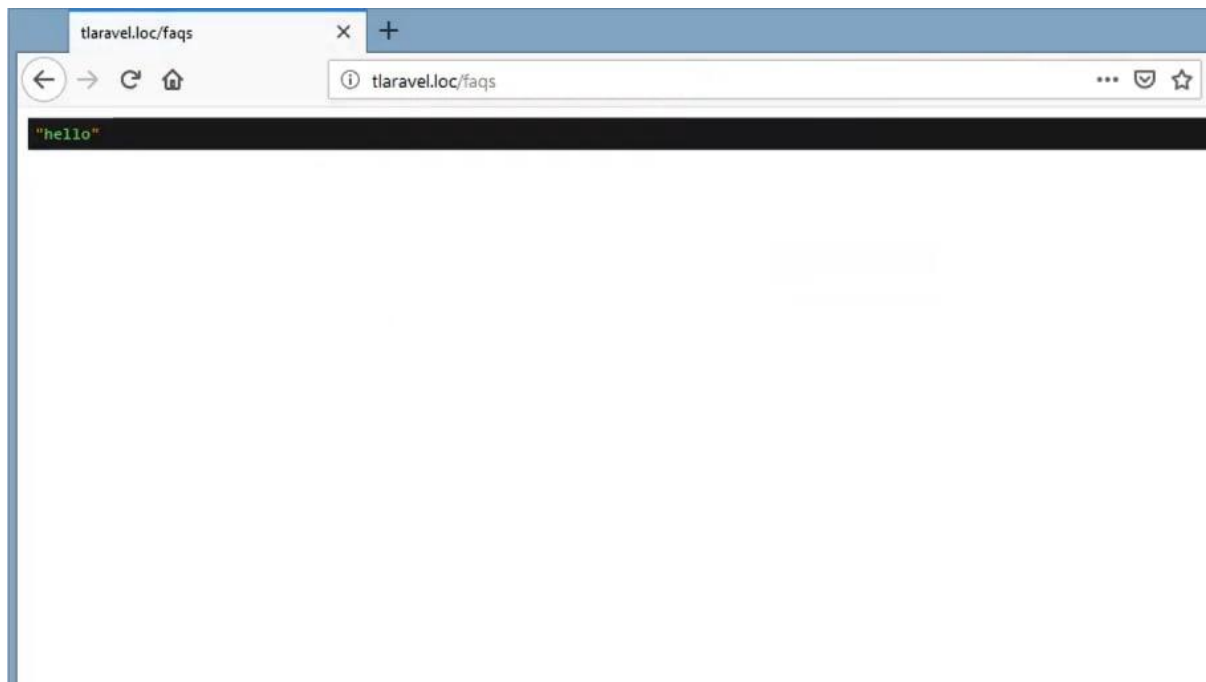
-також треба зробити перевірку:

```

30 $relativePath = "/" . $mod . "/" . $sub;
31 $routesPath = $path . $relativePath . "/Routes/web.php";
32
33 if(file_exists($routesPath)) {
34     Route::namespace("Modules\\" . $mod . "\\" . $sub . "Controllers")->group($routesPath);
35 }
36 }

```

-оновлюємо сторінку браузеру та бачимо надпис:



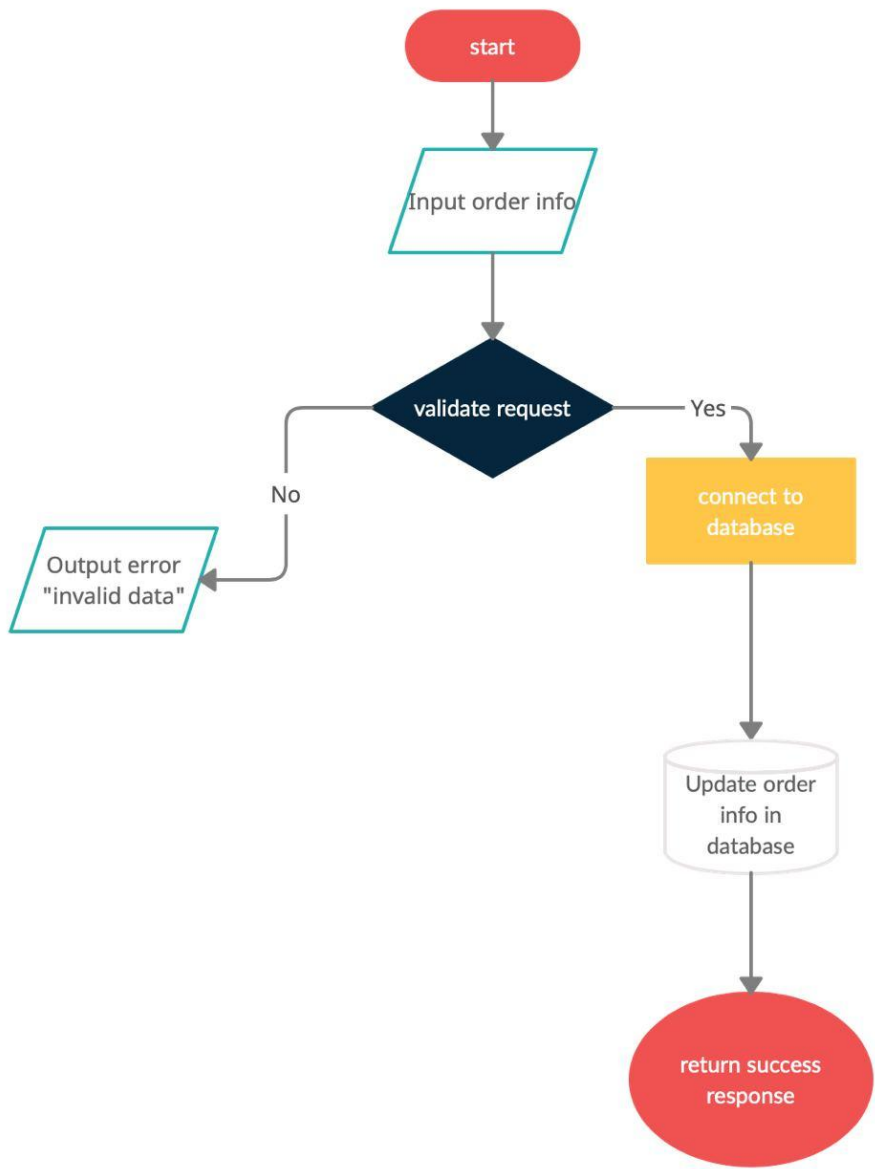
-цей надпис є відпрацюванням цієї строки, про яку я не розповів спочатку:

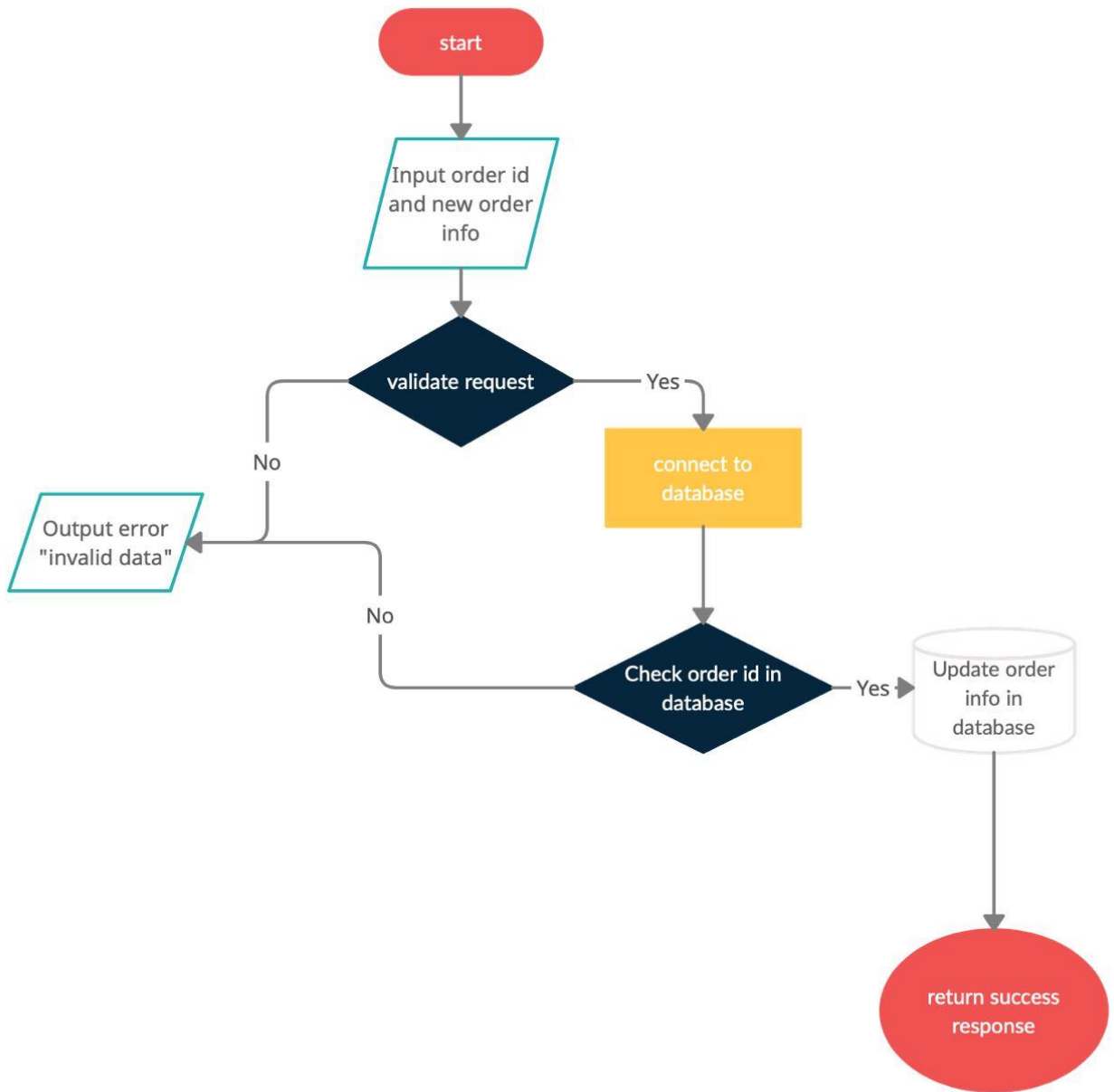
```
web.php x RouteServiceProvider.php x modular.php x Routes\web.php x Faq.php x FaqController.php x
\App\Modules\Admin\Faq\Controllers FaqController index()
<?php
/** Created by PhpStorm. ... */

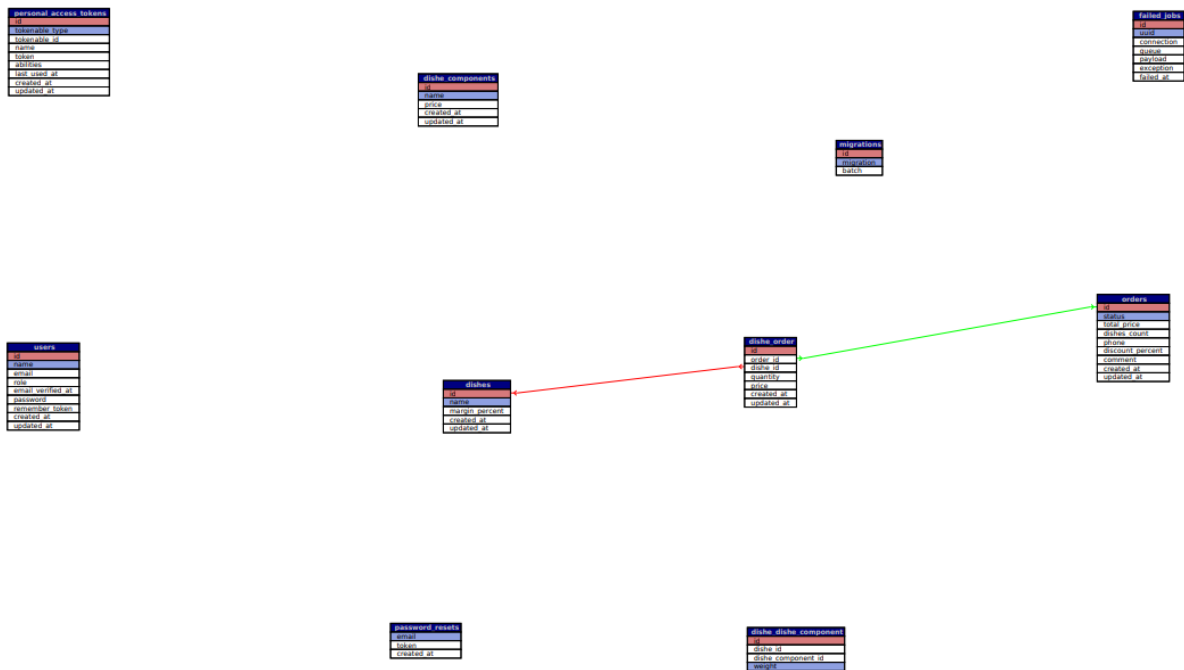
namespace App\Modules\Admin\Faq\Controllers;

use App\Http\Controllers\Controller;

class FaqController extends Controller
{
    public function index() {
        dd('hello');
    }
}
```







IV. Висновок з розділу.

Таким чином ми проаналізували:

- Різноманітніші пакети для нашого фреймворку, виявили їх особливості та переваги, чим саме вони допомагають та яку роботу облегшують.
- примітивний алгоритм створення модулів. Це було потрібно задля вивчення логіки та принципів модульного програмування. Я вважаю цей проект був дуже важливим для розуміння процесів. Розуміння у свою чергу є основою для успішного програмування, бо без розуміння не буде якісного кінцевого продукту. Розуміння допомагаю нам також створювати та використовувати нові підходи до різних питань, винайдення будь-яких новий методів та способів програмування.

Багатоцільовий аналіз функціоналу пакетів та модульних структур дав нам змогу зрозуміти більш детально види пакетів та логіку модульного проектування.

VI. Висновок

Ми дослідили питання автоматизації дуже детально з багатьох сторін. Ми знайшли переваги та недоліки автоматизації, її певний систем. Зробили детальний аналіз frameworks за допомоги порівняльної таблиці. Також ми описали певні модулі та шляхи їхньої реалізації. Висновок до кожного розділу допомагає нам усвідомити важливість та роль тієї чи іншої сторони даної роботи. Я вважаю, що автоматизація – великий крок вперед, який веде нас на шлях майбутнього, а ця робота змогла проаналізувати цей крок як з теоретичної, так і з практичної сторони.