

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

**Пояснювальна записка
до дипломної роботи**

бакалавр

(освітньо-кваліфікаційний рівень)

**на тему «Saas платформа обліку фінансів для малого та середнього
бізнесу»**

Виконав: студент 4 курсу, групи ПЗ-17д
напряму підготовки 121 „Інженерія
програмного забезпечення”

_____ Чепель С. Л.
(підпис)

Керівник,
доцент, к.т.н. _____ Іванов В.Г.
(підпис)

Рецензент,
доцент каф. ПМ,
• _____
(підпис)

СЄВЕРОДОНЕЦЬК – 2021

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
бакалаврської роботи студента гр. ІПЗ-17д Чепеля С.Л.

Науковий керівник
доцент, к.т.н. _____ Іванов .В.Г
ПБ, посада

Оцінка наукового керівника: _____

Рецензент _____ . каф. ПМ СЧУ ім. В. Даля
ПБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту: _____

Голова ЕК,
проф. кафедри ПМ
д.т.н., доцент

підпис

Захожай О.І.

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ**

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 121 „Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____ Лифар В.О.

«___» _____ 2021 р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Чепеля Сергія Леонідовича

1. Тема роботи Saas платформа обліку фінансів для малого та середнього бізнесу.

керівник роботи к.т.н., доцент Іванов Віталій Геннадійович

затверджені наказом вищого навчального закладу від «22» квітня 2021 року

№ 68/14.04

2. Строк подання студентом роботи 12 червня 2021 р.

3. Вихідні дані до роботи

Об'єктом досліджень є розробка веб додатку з обліку фінансів для малого та середнього бізнесу.

3.1 Літературні джерела:

Переваги побудови динамічних сайтів - hackernoon.com [Електронний ресурс] - Режим доступу: <https://hackernoon.com/the-advantages-ofbuilding-dynamic-websites-bc914071a155>

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналіз предметної галузі (огляд літератури), з висвітленням наступних питань:

Основні завдання веб додатку.

Аналіз конкурентів.

4.3 Основна частина, в якій висвітлити:

Вибір і опис методології проектування.

Технології розробки сайту.

Вибір засобів розробки.

Проектування схеми і опис бази даних.

Вибір СУБД.

Проектування макету сайту

Реалізація веб сайту.

4.4 Висновки

4.4 Перелік використаних джерел

5. Перелік графічного матеріалу немає

6. Дата видачі завдання 23 квітня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	23.04.21	
2	Укладання і погодження з керівником плану і етапів виконання роботи	23.04.21	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	26.04.21	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	02.05.21	
5	Розробка технічного завдання	09.05. 21	
6	Укладання програмного продукту	11.05.21	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	25.05.21	
8	Здача готової пояснювальної записки на кафедрі	06.06.21	
9	Укладання доповіді і презентації	07.06.21	

Студент

_____ Чепель С. Л..
(підпис)

Керівник роботи

_____ Іванов В. Г..
(підпис)

РЕФЕРАТ

Текст – 65., рис. – 30, літературних джерел – 16

Фінанси є основною складовою будь-якого малого та середнього бізнесу. Уміння правильно підраховувати та розподіляти прибуток та витрати є також невід'ємною частиною процесу ведення бізнесу. На жаль, не завжди є час або потрібні люди з навичками правильного ведення фінансів. Тому питання розробки веб додатка для автоматизованого обліку фінансів є актуальним.

Метою роботи є розробка платформи для обліку фінансів для малого та середнього бізнесу

Ключові слова: веб додаток, фінанси, Laravel, VUEJS

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	9
1.1. Основні поняття.....	9
1.2. Огляд існуючих рішень.....	12
1.3. ВИСНОВОК ДО РОЗДІЛУ 1.....	23
РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ	24
2.1. Порівняння РНР фреймворків.....	24
2.2. Фреймворк Ларавель	32
2.3. ВИСНОВОК ДО РОЗДІЛУ 2.....	41
РОЗДІЛ 3. РОЗРОБКА ВЕБ ДОДАТКА.....	42
3.1. Постановка задачі.....	Ошибка! Закладка не определена.
3.2. ВИСНОВОК ДО РОЗДІЛУ 3.....	42
ВИСНОВКИ.....	59
СПИСОК ДЖЕРЕЛ.....	60

ВСТУП

Актуальність.

Фінанси є основною складовою будь-якого малого та середнього бізнесу. Уміння правильно підраховувати та розподіляти прибуток та витрати є також невід'ємною частиною процесу ведення бізнесу. На жаль, не завжди є час або потрібні люди з навичками правильного ведення фінансів. Тому питання розробки веб додатка для автоматизованого обліку фінансів є актуальним.

Об'єкт дослідження – створення веб платформи.

Предмет дослідження – розроблення веб платформи для обліку фінансів для малого та середнього бізнесу

Мета дослідження: розробити веб платформу для обліку фінансів для малого та середнього бізнесу.

Задачі дослідження:

- 1) Проаналізувати предметну галузь обраної області.
- 2) Визначити існуючі підходи та технології для створення веб застосунків.
- 3) Проаналізувати інструменти розробника для створення веб застосунків.
- 4) Порівняти інструменти за якісними та кількісними характеристиками.
- 5) Реалізувати веб застосування

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Основні поняття

Будь-яка компанія веде фінансовий облік. Зрозуміло, робить це вона не просто так і навіть не через те, що цього вимагають контролюючі органи. Фірмі це необхідно, компанії створюються з метою отримання прибутку. Також і родини можуть вести власний фінансовий облік для контролю і створення тих чи інших цілей. Вести облік зовсім не настільки складно. Як і кожна звичка, ведення особистих фінансів повинно здійснюватися з власного бажання та з власною внутрішньою мотивацією. Спершу ця справа видається важкою і не підйомною, але насправді вже за місяць людина витрачає 5-7 хв. часу в день (а деколи й 1- 2 хв.) для обліку власних фінансів, а згодом на основі зібраної статистики може ввести корективи в некорисні чи занадто витратні звички або ж попрацювати над збільшенням джерел доходу.

Бізнес фінанси не дадуть відповідь на запитання «де взяти гроші?», але вони можуть допомогти з кращим розумінням звідки у вас появляються кошти й куди вони витрачаються, а вже на основі цієї інформації кожен будує власні висновки.

Корисність ведення фінансів – це перш за все розуміння того як і за що ми живемо, правильні висновки від ведення фінансів – жити відповідно до власних можливостей. Якщо людина хоче жити (витрачати більше) краще або ж заощаджувати більше – вона повинна заробляти більше або ж бюджетувати на перспективу з контролем «некорисних» витрат.

Підходи різні і кожен виділяє свою користь від ведення персональних фінансів. Деякі з плюсів ведення обліку власних доходів й витрат, які можуть слугувати для родини мотивацією: Коли людина почне вести фінансовий облік, її фінансовий стан може стабілізуватись або й покращитись (залежно від

«запущеності» поточного стану її коштів). Людина дізнається багато нового

про рух власних коштів. Вона дізнається скільки грошей витрачає протягом місяця. Також дізнається, що насправді витрачає не таку суму коштів як їй здається (а швидше за все більше, ніж вона думає). У неї перед очима буде наглядна картина того, куди йдуть ваші гроші. Вона дізнається, які статті витрат з'їдають більшу частину грошей, а які – меншу. Вона почне тримати свої фінанси під контролем. Дуже багато хто цього не робить і тому не знає, куди зникають їхні гроші. Дивлячись на свій фінансовий звіт, людина почне замислюватися про цінність грошей: чи варто зайвий раз витратити гроші на чергову непотрібну імпульсивну покупку. Вона буде в курсі свого фінансового стану і буде знати як з часом він змінюється.

Існує багато програм, веб-сайтів, мультиплатформових рішень для ведення фінансів – гугл в допомогу.

Основні критерії вибору:

- зрозумілість та якість інтерфейсу (наявність якісного FAQ для вивчення програми),
- доступність потрібних вам функцій (звіти, таблиці, деталізація витрат тощо),
- синхронізація (можливо вам зручно вносити витрати з різних комп'ютерів чи телефонів, треба мати доступ до бази даних з різних місць),
- хороша репутація і відгуки та підтримка продукту його автором (виправлення багів, апдейти тощо).

Основні можливості комп'ютерного ведення сімейного бюджету:

1) Формування звітів. Це дуже важлива частина ведення сімейного бюджету. Комп'ютер дозволяє швидко сформувати потрібні родині звіти для подальшого аналізу. Аналіз вашого бюджету дозволяє розробляти стратегію і виробляти план подальших дій, або вносити коригування в уже існуючий план.

2) Ведення декількох типів рахунків. Людина може вести кілька рахунків. Наприклад, сімейний бюджет, депозит в банку і т.д. Вся інформація

відображається на екрані ПК, що дуже зручно.

3) Планування персональних фінансів. Відмінна можливість вносити майбутні платежі і передбачувани витрати. Планування дозволить людині розраховувати свої дії на кілька кроків вперед.

4) Розрахунок кредитів і вкладів. Багато програм мають вбудовані калькулятори для розрахунку кредиту / вкладу. Особливо зручні ті, що дають можливість розрахунку закладами. Дуже зручна і мотивуюча річ.

5) Контроль за боргами. Є можливість додавати своїх боржників, для точного відображення персональних фінансових справ. Ця функція також допомагає не забути кому і скільки людина зайняла, адже іноді вона може просто забути кому позичала гроші. А боржники, в свою чергу, забували у кого вони займали.

6) Захист даних. Всі свої дані людина зможе приховати від зайвих очей надійним паролем.

7) Нагадування про необхідні платежі. Заплатити за квартиру, погасити кредит, повернути борг, купити подарунок - щоб нічого не забути, програма по веденню сімейного бюджету нагадає про це. Відмінна функція - процесор комп'ютера має куди кращу пам'ять, ніж людина.

Як вести сімейний бюджет і його облік

Почнемо з того, що облік бюджету необхідно вести якомога частіше. Бажано щодня. Це займає зовсім небагато часу. Все, що від вас вимагається на перших порах - це враховувати всі свої доходи і витрати і заносити їх в комірну книгу, або спеціальну програму. Ведення сімейного бюджету. На те, щоб зібрати дані для аналізу доходів і витрат, бажано витратити не менше 2-3 місяців. Зібрані дані за місяць, швидше за все, не дадуть об'єктивного уявлення про сімейний бюджет.

Тому, якщо ви ще не вели облік бюджету або ведете його нерегулярно, то змусьте себе провести цей експеримент протягом 2-3 місяців. Найскладніше, це облік витрат. Великі витрати запам'ятати нескладно. А ось дрібні, швидше за все ви будете забувати часто.

Саме ці дрібні витрати, в кінці місяця, складуть ті самі 10-30%, які витікають через "дірки ваших кишень". Дірами є ваша пам'ять, неуважність, легковажність.

1.2. Огляд існуючих рішень

Найперші додатки і сервіси з обліку фінансів просто замінювали собою старі добрі блокноти: дозволяли вести облік не на папері, а в комп'ютері. Але можливостей у електронного обліку більше, так що незабаром у таких сервісів стала розширюватися функціональність: з'явилася ієрархічна система категорій витрат і доходів, нагадування про оплату комунальних платежів або про повернення боргу. Додалися аналітичні інструменти, планування бюджету, можливість вести облік спільно з іншими користувачами і так далі. Багато додатків стали крос-платформеними. Але навіть в самих просунутих з них облік вимагав ручної роботи: користувач повинен був самостійно вказувати суму і категорію кожної покупки або доходу. Тому наступним етапом розвитку додатків для обліку фінансів стала автоматизація введення даних.

Автоматизації багато в чому допомогло розвиток технологій електронних платежів і, як не дивно, нормативно-правової бази. У будь-якому банку тепер можна отримувати СМС-повідомлення про покупках з карти, і програми обліку фінансів навчилися обробляти такі повідомлення. Деякі додатки навіть підтримують пряму інтеграцію з банками і прив'язку до рахунків. В силу того, що люди розраховуються картами все більше, значно зменшується кількість даних про покупки, які доводиться вносити вручну. Крім того, податкова служба зробила для магазинів обов'язкової відправку чеків в ФНС - і тепер на кожному чеку є QR-код, за допомогою якого можна отримати електронний чек на свій смартфон. Це ще більше автоматизує введення даних - в програму завантажується не тільки загальна сума

покупок, але і вартість кожного товару. Причому немає різниці, яким методом користувач розраховувався в магазині, - готівкою або карткою.

Переведення більшої частини бізнес-процесів в онлайн вбачається невідворотним у перспективі найближчих декількох років. Уже зараз це торкнулося багатьох областей роботи підприємств, і цілком логічно, що насамперед необхідно автоматизувати найбільш рутинні завдання.

Якщо говорити про малий бізнес, зокрема про інтернет-магазини, важко навести приклад чогось більш рутинного, але абсолютно необхідного для нормальної роботи, ніж ведення бухгалтерії. І зараз технології вже достатньо розвинені, щоб можна було автоматизувати цей напрямок за допомогою сервісів онлайн-бухгалтерії, а час і гроші, які вивільняться в результаті, направити на розвиток свого бізнесу.

Більшість власників малого бізнесу щодня вирішують безліч завдань, від яких залежить нормальна робота компанії. І можна бути впевненим, що бухгалтерський облік і бухгалтерська справа знаходяться далеко не в верхній частині цього списку з погляду задоволення від виконання таких занять.

Переваги ведення бухгалтерії онлайн

У самій природі підприємництва закладена необхідність отримувати переваги з будь-якої діяльності. Наприклад, можливість заощадити час, отримати більшу ефективність від зайнятості в певному напрямку або ж використовувати будь-який інструмент для того, щоб зробити управління бізнесом простішим?

У 2016 році ви можете отримати всі перераховані переваги за рахунок впровадження всього лише однієї зміни – почати вести бухгалтерський облік онлайн. І що важливо – на ринку доступний вибір компаній, які пропонують послуги, як то кажуть, на будь-який смак і гаманець.

Але цим справа не обмежується. Ось ще ряд позитивних моментів, які можуть бути вагомим аргументом на користь використання таких сервісів.

Доступ у будь-якому місці і в будь-який час

При використанні хмарних бухгалтерських рішень ви вже не прив'язані до одного комп'ютера, наприклад, у своєму офісі. Дані розміщені на віддалених серверах або в «хмарі», а деякі сервіси також надають своїм користувачам мобільні додатки-клієнти. Таким чином, ви можете отримати потрібну інформацію про свій бізнес завжди і всюди – за допомогою веб-браузера або через мобільний додаток.

Централізоване оновлення

Практично будь-яке програмне забезпечення, особливо якщо йдеться про професійний софт, потребує регулярного оновлення та виправлення помилок. Якщо говорити про настільну версію, встановлену на комп'ютер, цим повинен займатися співробітник компанії. У випадку з веб-версією все оновлення здійснюється постачальником послуг – ви зможете використовувати найновішу версію програмного забезпечення без будь-яких додаткових витрат.

Безпека і збереження даних

Якщо вся документація зберігається в офісі, достатньо пролити чашку кави на стопку паперів або квитанцій, щоб втратити всю цінну інформацію за останні півроку. Та й старі добрі офісні комп'ютери іноді ламаються, часом, що прикро, через банальні перепади напруги.

У випадку з онлайн-бухгалтерією всі дані надійно зберігаються в хмарі, вони надійно захищені на боці сервера, адже їхня безпека забезпечується розробниками, які часом витрачають десятки тисяч доларів, інвестуючи їх у безпеку.

Крім того, ці сервіси використовують таку корисну штуку, як резервне копіювання – збереження даних забезпечується їхнім регулярним дублюванням в іншому місці, що дозволяє відновити інформацію з резерву в разі будь-яких збоїв.

Одна з найбільших переваг використання онлайн-бухгалтерії – зниження витрат власника бізнесу. Для невеликої фізичної або юридичної особи, яка веде свій бізнес одноосібно, наймання бухгалтера може бути занадто дорогим задоволенням. Навіть просто придбання ліцензії на програмне забезпечення та його установка на домашній комп'ютер може відчутно вдарити по бюджету.

Практично всі інтернет-сервіси, які дають можливість ведення бухгалтерського обліку онлайн, працюють за схемою підписки. Тобто ви просто платите відносно невелику суму щомісяця протягом усього часу, поки користуєтеся можливостями системи.

Деякі рішення для корпоративних клієнтів дають своїм користувачам ще й таку корисну можливість, як електронна звітність. Створювати документи і відправляти до відповідних державних органів можна в будь-який час і без необхідності відвідувати їх особисто, простоювати в черзі та інших незручностей.

Інформація за різними рахунками, виписки, звіти та інше – все це створюється в автоматичному режимі або може бути створене одним натисканням кнопки. Відсутність необхідності заповнювати такі документи вручну, крім того, виключає можливість помилки.

Ведення бухгалтерії – заняття досить-таки втомливе, через що працівників, які цим займаються, складно назвати високопродуктивними. Якщо робити це через Інтернет, із використанням автоматизованих систем, робота людей

значно спрощується. Наприклад, немає потреби зайвий раз заповнювати папери, адже практично половина роботи вже зроблена.

Сервіси онлайн-бухгалтерії в Україні

У нашій країні використання сучасних технологій для управління бізнесом стає все популярнішою практикою. Правда, якщо говорити про онлайн-бухгалтерію, список сервісів поки недостатньо широкий і складно назвати рішення, яке було б по-справжньому універсальним. Ми зробили для вас добірку тих, на які дійсно варто звернути увагу.

Taxer.ua

TAXER онлайн-бухгалтерия для малого бизнеса 0 800 509 545 Напишите письмо [Вход](#) [РЕГИСТРАЦИЯ](#) РУС УКР

[ВОЗМОЖНОСТИ](#) [ЦЕНА](#) [Налоговый календарь](#) [Реквизиты](#) [Бланки](#) [База знаний](#) [Блог](#)









Быть предпринимателем — просто!

Таксер поможет вам:

- Сократить расходы на бухгалтерию
- Вовремя и без ошибок сдавать отчеты
- Защитить данные от взлома и изъятия
- Получить консультацию и помощь

[ПОПРОБОВАТЬ СЕЙЧАС](#) или [УЗНАТЬ О ВОЗМОЖНОСТЯХ](#)

ПРЕДПРИНИМАТЕЛЯМ **БУХГАЛТЕРАМ**

<ul style="list-style-type: none">  Бесплатная электронная отчетность в налоговую и пенсионный фонд  Автоматическое заполнение отчетных документов  Мастер формирования первичных документов  Автоматический расчет курсовой разницы 	<ul style="list-style-type: none">  Налоговый календарь с уведомлениями  Загрузка собственных шаблонов документов  Онлайн-оплата ЕН и ЕСВ  Юридические и бухгалтерские консультации
---	--

Сервіс орієнтований як на бухгалтерів, так і на підприємців, зареєстрованих як ФОП, що ведуть свою бухгалтерію самостійно. Він дозволяє відчутно скоротити витрати на бухгалтерію, а також надає підприємцям і бухгалтерам такі переваги, як:

регулярна відправка звітів до податкової служби та Пенсійного фонду;

покроковий майстер формування первинних документів;

податковий календар із повідомленнями;

автоматичний розрахунок різниці курсів;

можливість завантажувати власні шаблони документів;

та інші корисні функції.

Творці Taxer.ua добре попрацювали над базою знань із бухгалтерських і суміжних юридичних питань, а при необхідності можна отримати консультації у служби підтримки абсолютно безкоштовно.

Тарифна сітка як така тут відсутня, і всі можливості сервісу доступні користувачам за єдиним тарифом – підпискою, вартістю 450 гривень на рік.

MoneyPenny.me

Блог Служба підтримки Увійти Мова

Головна Характеристики Ціни Можливості

Безкоштовна версія

Працюйте ефективніше!

Отримуйте платню вчасно!

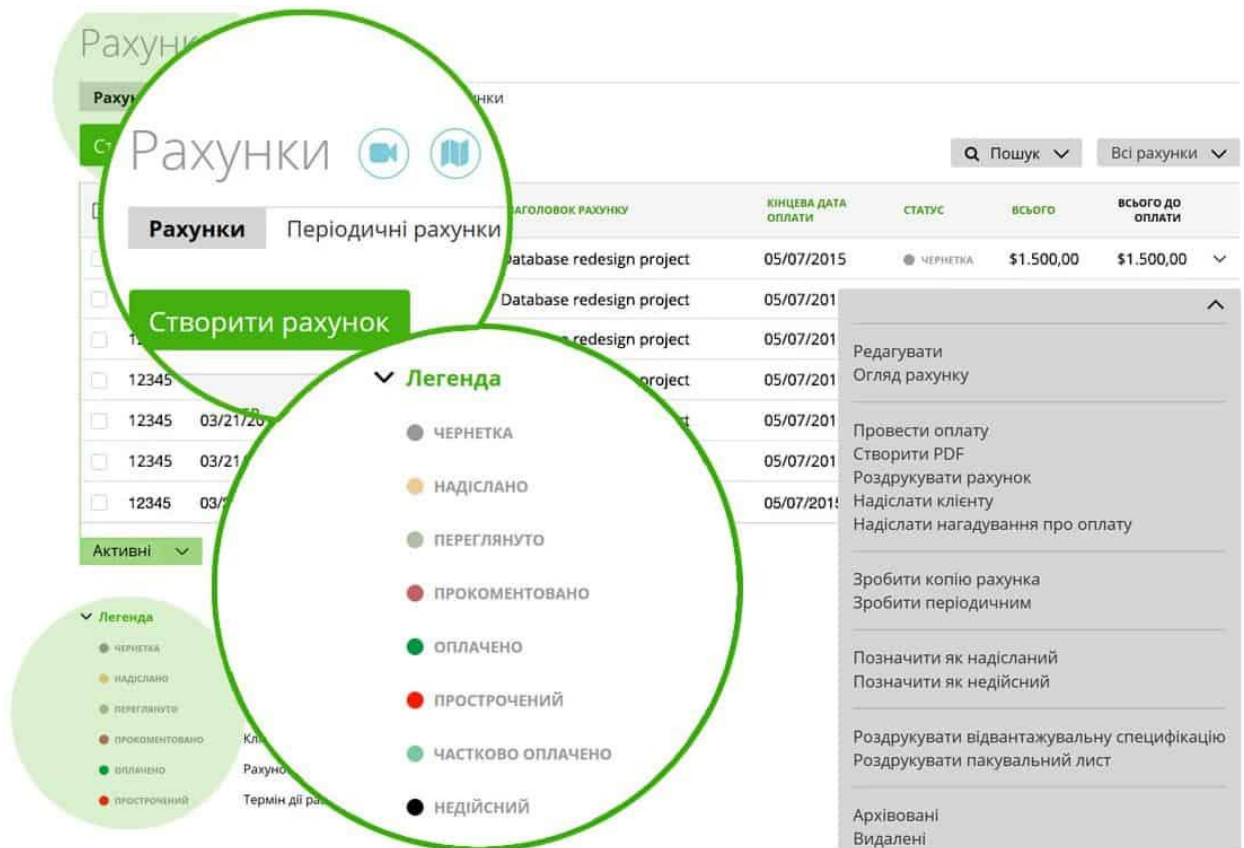
MoneyPenny - УНІВЕРСАЛЬНЕ білінгове програмне забезпечення, з яким ви легко зможете оптимізувати управління бізнесом і діловодством, удосконалити облік робочого часу та виставити рахунок-фактуру онлайн. Автоматизована білінгова система розроблена для беззаперечної перемоги власників малого бізнесу та фрілансерів у боротьбі зі звітністю та контролем. Гарантуємо зростання продуктивності без зусиль!

Спробувати безкоштовно

Повноцінний програмний комплекс для автоматизації управління бізнесом і діловодства, який буде однаково зручний як для власників малого бізнесу, так і для фрілансерів, які працюють як незалежні фахівці.

Ведення бухгалтерії на підприємстві та виставлення рахунків-фактур – одна з функцій цієї програми. Дані про витрати зберігаються в хмарі, а за допомогою мобільного додатку доступні і в будь-якому місці за умови наявності підключення до Інтернету. У цілому ж, як уже було написано

спочатку, можливості сервісу набагато ширші, докладніше з усім спектром напрямків можна ознайомитися на цій сторінці.



Сам продукт – стартап, орієнтований на глобальний ринок, і до якості реалізації розробники підійшли ґрунтовно. Приємно радує те, що доступна локалізація інтерфейсу українською мовою. Тарифні плани ви можете побачити нижче:

	НАЙПОПУЛЯРНИШИЙ	
<p>Стартовий</p> <p>Ідеально підійде для соло-фрілансера і подальшого використання безкоштовної пробної версії, допоможе звикнути до великої кількості бізнес-інструментів MoneyPenny.</p> <p>300€</p> <p>За місяць / для одного користувача</p> <p>Підписатись</p>	<p>Професійний</p> <p>Цей тариф повністю задовільнить потреби невеликих команд або ж фрілансерів, що планують аутсорсити свою роботу та самостійно контролювати всі фінансові операції.</p> <p>700€</p> <p>За місяць / для 5 користувачів</p> <p>Підписатись</p>	<p>Підприємницький</p> <p>Прогресивний, універсальний та зручний у використанні тарифний план спеціально створений для великих команд, що співпрацюють зі значною кількістю клієнтів і замовників.</p> <p>1500€</p> <p>За місяць / для 40 користувачів *</p> <p>Підписатись</p>

iFin.ua



Рус Ukr

[Отчетность](#)[Бухгалтерия](#)[Как начать](#)[Цены](#)[Безопасность](#)[Вопросы и ответы](#)Поддержка: [в регионе](#) (044) 22-77-0-66 (093) 17-00-8-66 (067) 35-46-4-02[Вход в систему](#)

Сдавать
электронные документы
теперь просто

**iFin Zvit это:****ЭЛЕКТРОННАЯ ОТЧЕТНОСТЬ**

- Все отчеты в Фискальную службу, ПФУ и Статистику
- Проверка отчетности на соответствие стандартам
- Качественная техническая поддержка

НАЛОГОВЫЕ НАКЛАДНЫЕ

Сдавайте и делитесь налоговыми накладными с вашими контрагентами

БЕСПЛАТНО СДАВАЙТЕ

- Налоговые накладные
- Отчет ЕСВ
- 4 отчета в ПФУ и Статистику

Узнать больше о бесплатных услугах

(044) 22-77-0-66 (093) 17-00-8-66 (067) 35-46-4-02

[Попробовать сейчас](#)

Один із перших в Україні сервісів для створення і відправлення електронної звітності, який працює ще з 2012 року. За допомогою iFin можна вести документообіг без необхідності заглиблюватися в нюанси бухгалтерської справи, відправляти звіти в Фіскальну службу і Пенсійний фонд в режимі онлайн, без необхідності навіть виходити з дому.

Крім перерахованих уже можливостей, користувачеві сервісу доступний такий функціонал, як механізм автозаповнення з усіма реквізитами, електронний документообіг з партнерами, автоматизований розрахунок зарплати співробітників та інших витрат на персонал.

1С: Підприємство (в хмарі)

Головна Додатки Переваги Запитання та відповіді Відеоінструкції Тарифи [Укр](#) [Рус](#)

1С Оренда програм **Онлайн**

Логін Пароль [увійти](#)

Чужий комп'ютер [Реєстрація](#) [Відновити пароль](#)

1С:Підприємство 8 через Інтернет

Ведіть потрібне Вам число підприємств в одній або різних базах даних в конфігураціях:

- 1С:Управління невеликою фірмою для України.
Комплексне готове рішення для обліку на підприємствах малого та середнього бізнесу.
- 1С:Бухгалтерія 2.0 для України (beta)

Для клієнтів ПриватБанку включено інтеграцію з «Приват24 для бізнесу»:

- платежі, виписки, зарплатний проект;
- електронний документообіг (Замовлення, Рахунок, Акт, Накладна з ЕЦП) і реєстрація податкових накладних в ЄРПН;
- курси валют і т.д.

[ЗАРЕЄСТРУВАТИСЯ](#) ▶

Настільна версія цієї програми – найпопулярніший продукт такого роду на території Росії і України. Мабуть, творці 1С зрозуміли, що далеко не для всіх ціна десктопної ліцензії доступна або, цілком можливо, пішли за загальним трендом, але тепер обидва продукти компанії – “1С: Управління невеликою фірмою для України” і “1С: Бухгалтерія 2.0 для України” – доступні як онлайн-сервіс.

Для клієнтів ПриватБанку є приємний бонус – інтеграція з «Приват24 для бізнесу», що істотно розширює можливості з управління підприємством і ведення бухгалтерії. Тариф за 1 робоче місце становить фіксовану суму – 199 грн/міс.

1С Онлайн

Тарифы	199 грн/мес
Количество доступных приложений (УНФ, Бухгалтерия...)	Без ограничений
Количество организаций и ФЛП	Любое (без ограничений)
Количество информационных баз	Без ограничений
Количество одновременно работающих пользователей	1
Подключение дополнительных рабочих мест	Без ограничений

Тариф	1 мес.**
1 рабочее место	199 грн.
Дополнительные рабочие места (цена 1 рабочего места)	199 грн.

Приват24 для бізнесу

Сервіс Приват24 знайомий багатьом із наших читачів, адже сам ПриватБанк – найбільший банк України, а в плані електронних операцій із грошима – ще й найбільш інноваційним з усіх.

Приват24 для бізнесу – це, по суті, той же інтернет-банкінг, але для корпоративних клієнтів. Якщо говорити про бухгалтерські операції, то він надає такі можливості, як:

цілодобовий доступ до рахунків та контроль за їхнім станом;

можливість перераховувати зарплати та інші виплати;

операції з обміну валют;

створення електронних звітів та їхнє відправлення до державних органів, а також інші можливості.

Сервіс – не повнофункціональне рішення для бухгалтерів, але добре підходить як додатковий інструмент для виконання цілого ряду завдань.

Підвищуйте ефективність бізнесу!

На початку цієї статті вже описувалися переваги ведення бухгалтерії та документообігу онлайн. Таким чином, ви стаєте незалежними від програмного забезпечення, встановленого на офісному або домашньому комп'ютері. Через Інтернет отримати доступ до актуальної інформації щодо фінансового стану вашого бізнесу можна в будь-якому місці, де б ви не перебували.

При використанні сервісів, навіть якщо з комп'ютером щось трапиться або ваш ноутбук буде вкрадений, ніхто не зможе заволодіти конфіденційною інформацією, адже дані зберігаються не на жорсткому диску.

Якщо ви хочете, щоб бізнес працював швидше й ефективніше, використання онлайн-сервісів для ведення бухгалтерії може бути дійсно вдалою інвестицією. Робота в хмарі знімає безліч обмежень, обходиться дешевше і в цілому підвищує якість роботи всередині команди.

1.3. ВИСНОВОК ДО РОЗДІЛУ 1

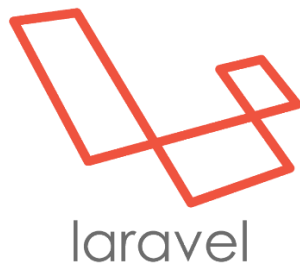
РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ

2.1. Порівняння PHP фреймворків

Десятки років PHP був найулюбленішим мовою програмування в світі. І це не випадково. PHP-розробка ведеться швидко, що виходить в результаті проекти відрізняються високим рівнем безпеки, їх легко підтримувати. Крім того, в ході розробки PHP-проектів доводиться писати не дуже багато власного коду завдяки тому, що існує величезна кількість PHP-бібліотек. В наші дні PHP використовується приблизно на 79% веб-сайтів.

Розробникам в справі створення їх чудових проектів допомагають різні PHP-фреймворки. Але що це за фреймворки? Чому програмісти вибирають саме їх? Зараз ми спробуємо в цьому розібратися і заодно з'ясуємо те, які саме фреймворки найкраще підходять для веб-проектів різних видів.

1. Laravel



Laravel - це безкоштовний опенсорсний PHP-фреймворк, який з'явився в 2011 році. З тих пір він зміг стати тим фреймворком, який найчастіше вибирають веб-розробники. Одна з головних причин цього полягає в тому, що Laravel

дозволяє розробляти складні веб-додатки простіше, швидше і безпечніше, ніж будь-які інші фреймворки.

Використання Laravel, крім того, полегшує вирішення багатьох великих завдань, що веде до того, що розробникам доводиться писати, у відповідних ситуаціях, менше коду. Серед таких завдань - маршрутизація, кешування, авторизація, аутентифікація. Завдяки використанню цього фреймворка спрощуються такі завдання, як перевірка електронних адрес користувачів, хешування і скидання паролів. Laravel вибирають, в основному, розробники B2B-рішень.

А після появи Homestead - заздалегідь підготовленого контейнера Vagrant, установка Laravel стала до крайності простий. Найсвіжіша на даний момент версія Laravel - 8 - вийшла у вересні 2020 року.

2. Yii



Yii - це простий, але високопродуктивний універсальний фреймворк, заснований на компонентній структурі. Він відомий завдяки високій продуктивності, але, в першу чергу, він знаменитий своєю простотою. Цей фреймворк з'явився в грудні 2008 року. Він дозволяє використовувати код сторонніх розробників, а наявний в ньому генератор коду Gii дозволяє швидко створювати базові структури, на основі яких можна будувати власні рішення.

Його творці дуже серйозно підійшли до питань безпеки. Зокрема, в нього вбудовані механізми хешування паролів, засновані на bcrypt, і засоби шифрування. Цей фреймворк відмінно підійде тим, хто хоче найняти розробника з Індії для роботи над проектами з сфер електронної комерції, створення CMS або форумів.

3. CodeIgniter



CodeIgniter - це фреймворк, який відомий тим, що для його приведення в робочий стан потрібно мінімальну кількість налаштувань. Це дозволяє тим, хто його обирає, працювати в хорошому темпі. Його багато разів, з моменту появи в 2006 році, оновлювали. Зараз найсвіжішою його версією є 4.0.3.

Розмір фреймворка, включаючи документацію, не перевищує 2 Мб, але він, незважаючи на таку компактність, пропонує розробникам безліч стандартних модулів для створення компонентів, що відрізняються надійністю і підходять для багаторазового використання. Це робить CodeIgniter відмінним вибором для тих, хто створює динамічні веб-сайти.

Цей фреймворк заохочує використання архітектури MVC, але він, в плані архітектури створюваних на його основі проектів, досить гнучкий, і не обмежує розробників лише цієї архітектурою. У ньому є надійна система захисту контенту, він володіє стандартними механізмами для захисту від CSRF- і XSS-атак.

Розробнику, придивляються до CodeIgniter, варто знати про те, що це одна з тих PHP-платформ, які найлегше освоїти. Його, крім того, дуже легко встановлювати. Ці фактори роблять його ідеальним вибором для новачків.

4. Symfony



Symfony вигідно відрізняється від інших PHP-фреймворків надійністю і зрілістю. Цей фреймворк з'явився давно, в 2005 році, тобто - існує набагато довше, ніж більшість інших розглянутих нами інструментів. Він популярний завдяки відповідності веб-стандартам і шаблонами проектування PHP.

Він включає в себе вбудовану систему тестування і заснований на великому MVC-фреймворк. Це робить його ідеальним вибором для великомасштабних веб-проектів, що створюються на рівні організацій. Він, зокрема, підтримує роботу з різними базами даних.

Якщо ви збираєтеся найняти веб-розробника в Індії на Symfony-проект, вам варто поцікавитися досвідом розробника в сфері застосування цього фреймворка і його попередніми роботами. Справа в тому, що цей фреймворк відрізняється досить крутий кривий навчання. Але при цьому будь-який, хто майстерно володіє Symfony, може називатися справжнім майстром веб-розробки.

5. CakePHP



CakePHP - фреймворк, який існує з 2005 року, відомий тими зручностями, які він дає веб-розробникам. Він потребує досить скромних налаштувань, не вимагає користуватися XML- або YAML-файлами. У нього є власна ORM, що вигідно відрізняє його від інших подібних інструментів. У плані безпеки у нього теж все добре, зокрема, в ньому є система захисту від CSRF-атак.

Його вбудована підсистема, звана «Components and Helpers», полегшує роботу програмістів, позбавляючи від необхідності робити багато чого самостійно, те ж саме можна сказати і про набір його бібліотек, що реалізують багато корисних можливостей. Завдяки цьому його можна назвати виправданим вибором для веб-проектів, в яких реалізований якийсь рідкісний або нестандартний функціонал.

CakePHP був одним з найперших PHP-фреймворків, що підтримують MVC, які ввірвалися в світ веб-розробки. Він використовується в великих відомих проєктах на кшталт 10 Fast Fingers, Printivo, Visit NC і Coconala.

6. Zend Framework



Zend Framework - це ще один фреймворк, в основі якого лежать компоненти. Його ще називають «фреймворком проміжного рівня». Він з'явився в 2006 році, зараз найсучаснішою є його версія 3.0.0. У ньому застосовується об'єктно-орієнтована MVC-підхід, який відомий тим, що допомагає розробникам займатися своєю справою, ні на що не відволікаючись.

Він дозволяє завантажувати необхідні компоненти у вигляді окремих бібліотек. При цьому можна, без особливих складнощів, інтегрувати в проекти і зовнішні бібліотеки. При його використанні вкрай просто організувати багаторазове використання коду.

Правда, з огляду на складність цієї платформи, з нею може бути непросто розібратися, особливо тим, хто переходить на Zend Framework з Laravel або Symfony. До цього фреймворку зазвичай вдаються лише досвідчені розробники.

7. Phalcon



Phalcon - це особливий РНР-фреймворк, вихідний код якого написаний на мові С. Це може здатися дивним, але завдяки такому підходу даний

фреймворк можна назвати найшвидшим проектом такого роду. Робота з цим фреймворком не означає необхідності мати досвід С-розробки. Програміст, що вибирає Phalcon, може мати справу лише з класами і просторами імен PHP, що генеруються фреймворком.

Phalcon - це високопродуктивний фреймворк, що розміщується в пам'яті і охоплює всі рівні веб-розробки, який з'явився в кінці 2012 року. Те, що в його основі лежить скомпільований код, дуже добре позначається на його продуктивності. Навіть застосовується в ньому система шаблонізації, Volt, відома своєю високою швидкістю. У ньому є допоміжні класи.

Phalcon, крім швидкості, пропонує розробнику добре організовану систему управління ресурсами і універсальну автоматизовану систему завантаження ресурсів. В результаті цей фреймворк ідеально підійде тому, хто докладає зусиль щодо забезпечення надзвичайно швидкі веб-проекти.

8. FuelPHP



FuelPHP - це MVC-фреймворк, що вийшов в 2011 році і підтримуваний силами спільноти розробників. Він відомий своєю гнучкістю. У ньому реалізована особлива версія MVC - HMVC (Hierarchical Model-View-Controller - ієрархічна версія архітектури модель-вид-контролер). HMVC, на відміну від MVC, заохочує багаторазове використання коду. FuelPHP, крім

того, пропонує розробникам відмінну розширюваність, модульність, хорошу систему організації коду. В результаті цей фреймворк дозволяє програмістам економити час і дбайливо витрачає системні ресурси.

FuelPHP дозволяє створювати веб-проекти різних масштабів. Він відрізняється надійною і добре продуманою системою безпеки з підтримкою Output Encoding, із захистом від CSRF- і XSS-атак.

У ньому є унікальна утиліта командного рядка, але цим його корисні можливості не обмежуються. Серед них - вельми просунута вбудована ORM. Серед інших його можливостей - підтримка розробки RESTful-API, хороша система маршрутизації, вбудовані механізми захисту від вразливостей. Серед відомих веб-проектів, в яких використовується FuelPHP можна відзначити Front Desk - систему управління майном, використовувану в готельному бізнесі.

9. Slim



Slim - це, що відображено в його назві, мікро-фреймворк, що відрізняється компактністю і швидкістю. Причина таких його характеристик криється в тому, що він абсолютно незалежний від стороннього коду. Він був створений в 2010 році, на даний момент найсвіжіша його версія - це 4.5.0.

Це один з кращих фреймворків для початківців. Його дуже легко освоїти, він відрізняється доброзичливою і простою документацією. На нього варто поглянути тим, кого цікавить розробка RESTful-API. Він полегшує вирішення таких завдань, як маршрутизація, шифрування файлів cookie, HTTP-кешування на стороні клієнта.

Фреймворк Slim призначений для маленьких але потужних веб-додатків і API. Десятки років PHP був найулюбленішим мовою програмування в світі. І це не випадково. PHP-розробка ведеться швидко, що виходить в результаті проекти відрізняються високим рівнем безпеки, їх легко підтримувати. Крім того, в ході розробки PHP-проектів доводиться писати не дуже багато власного коду завдяки тому, що існує величезна кількість PHP-бібліотек. В наші дні PHP використовується приблизно на 79% веб-сайтів.

2.2. Фреймворк Ларавель

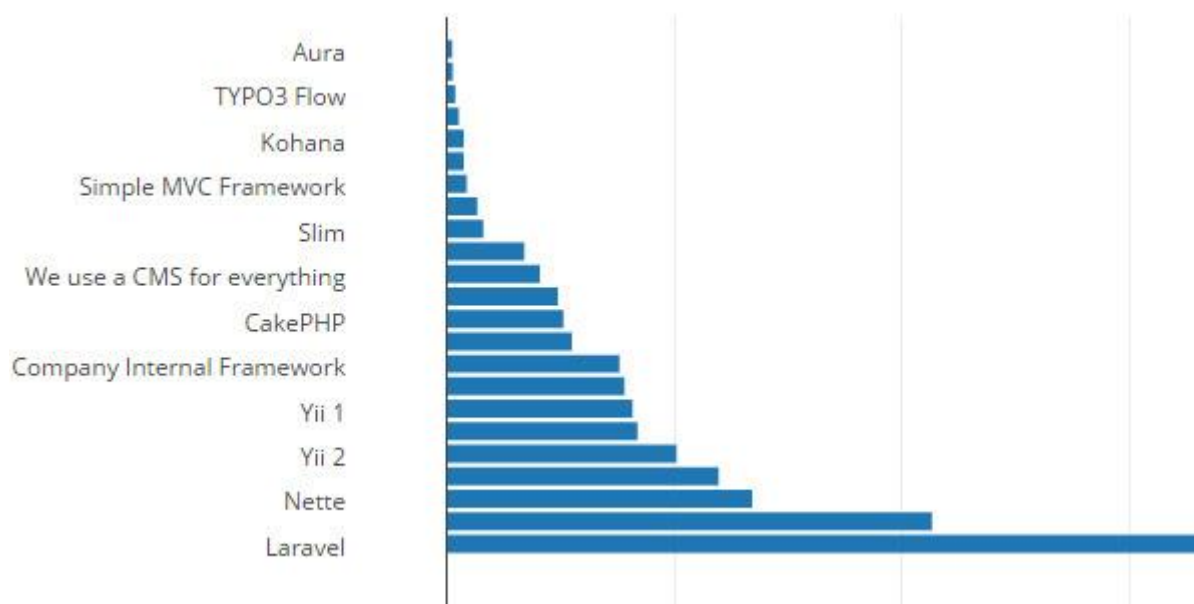
Laravel - це фреймворк для web-додатків з виразним і елегантним синтаксисом. Він дозволить спростити вирішення основних наболілих завдань, таких як аутентифікація, маршрутизація, сесії і кешування. Laravel - це спроба об'єднати все найкраще, що є в інших PHP фреймворк. Основні переваги Laravel:

- велика екосистема з миттєвим розгортанням своєї платформи. Офіційний сайт надає безліч мінлива і інформації для ознайомлення;
- документація Laravel близька до досконалості;
- у Laravel є свій движок для шаблонів Blade, «гарний» синтаксис мови, який сприяє вирішенню всіх необхідних завдань, таких як аутентифікація, сесії, кешування і маршрутизація RESTful.

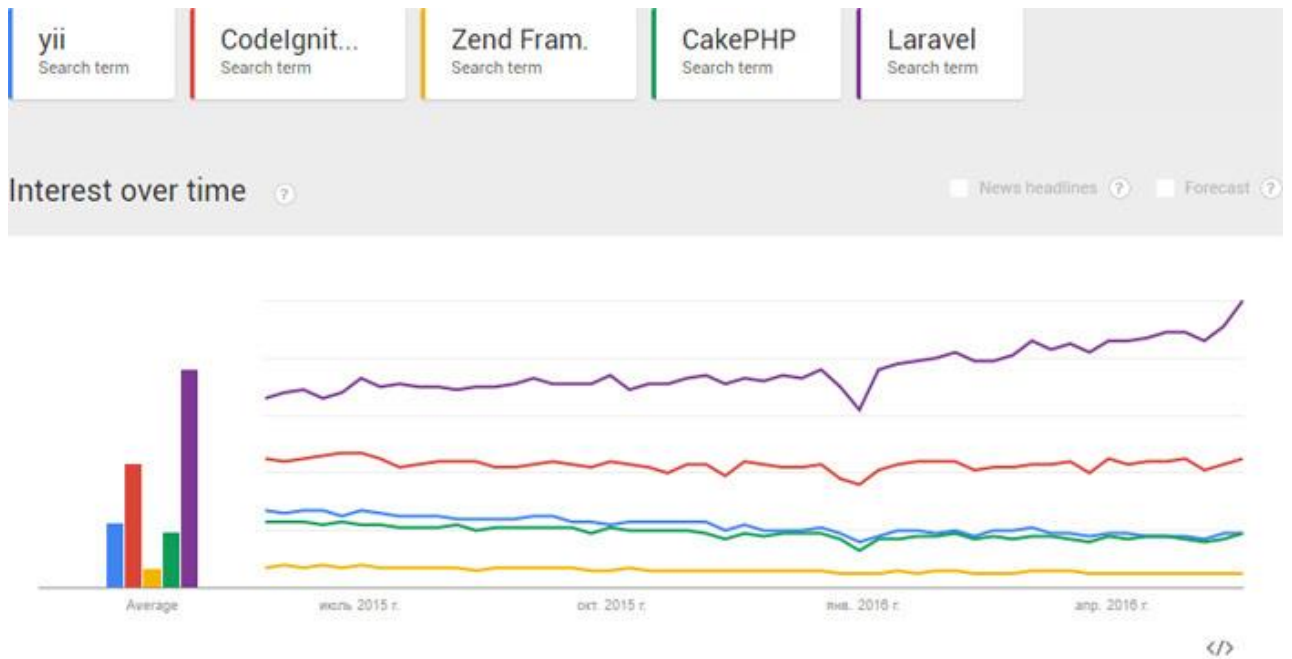
Незважаючи на свою молодість (перший реліз вийшов в 2011 році), це вже зовсім зрілий продукт, і, згідно з опитуванням, проведеним порталом SitePoint, він посідає перше місце за популярністю серед розробників на РНР.

Популярність фреймворка Laravel

Для того щоб оцінити популярність фреймворка, пропоную звернутися до декількох джерел статистики використання. Статистика популярності від Sitepoint за 2015 рік.

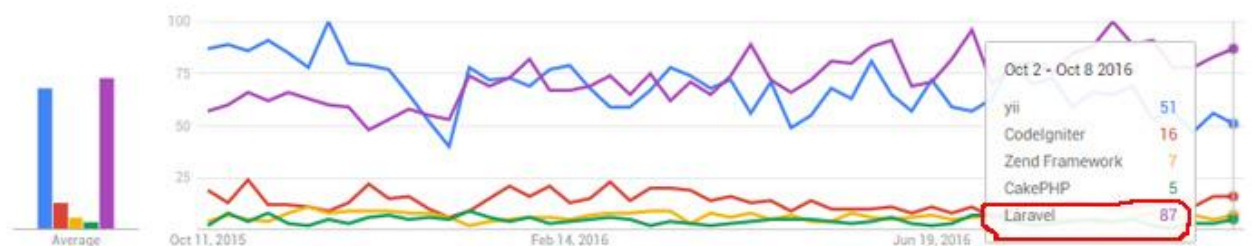


Популярність за статистикою запитів, яку надає Google Trends.



Як видно з графіків і статистики від SitePoint, а також зі статистикою Google Trends, фреймворк Laravel займає впевнене перше місце і постійно набирає популярність, не зменшуючи обертів. Що, загалом, дуже добре, враховуючи, що це загальносвітова статистика.

Якщо ми візьмемо статистику запитів в Google Trends по Росії, то картинка дещо зміниться:



Але з графіка очевидний факт, що протягом останніх 4 місяців фреймворк Laravel утримує позиції лідера і продовжує набирати популярність.

Це означає, що більшість нових проектів в країні тепер створюються з використанням активно-розвивається Laravel. З огляду на тренди, якщо ви давно хотіли почати вчити Laravel, то зараз той самий момент, коли це пора зробити.

Можливості Laravel

Пакети - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткові можливості вже доступні у вигляді таких модулів.

Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent.

Логіка програми - частина розробляється, оголошена або за допомогою контролерів, або маршрутів.

Зворотній маршрутизація пов'язує між собою генеруються додатком посилання і маршрути, дозволяючи змінювати останні з автоматичним оновленням пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL.

REST-контролери - додатковий шар для поділу логіки обробки GET- і POST-запитів HTTP.

Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключати файли їх визначень в include. Завантаження на вимогу запобігає завантаження непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються.

Укладачі уявлень - блоки коду, які виконуються при генерації уявлення (шаблону).

Інверсія управління - дозволяє отримувати екземпляри об'єктів за принципом зворотного управління. Також може використовуватися для створення і отримання об'єктів-одинаків.

Міграції - система управління версіями для баз даних. Дозволяє зв'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми.

Модульне тестування (юніт-тести) - грає дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання помилок.

Сторінковий висновок - спрощує генерацію сторінок, замінюючи різні способи вирішення цього завдання єдиним механізмом, вбудованим в Laravel.

особливості фреймворка

Laravel - це особливий фреймворк з сильним брендингом, тому майже все особливе в Laravel має унікальну назву.

Dotenv

Де ви зберігаєте секретну інформацію вашого застосування, таку як пароль від БД, логін для email та інші речі? Laravel 5 використовує добре відомий файл .env, який також використовують багато фреймворки для інших мов програмування. Це файл, в якому ви визначаєте пари ключ-значення для будь-якої вашої секретної інформації.

PSR-4

Laravel - перший фреймворк, що підтримує PSR-4. Прямо з коробки Composer автоматично завантажить всі класи з каталогу app, використовуючи стандарт автозавантаження PSR-4.

Це означає, що у вас може бути один простір імен для вашого застосування, і ви можете структурувати його, як захочете, і як вважаєте за логічним. Laravel не вимагає від вас розміщення певних файлів в певних папках.

IoC-контейнер - розширення ядра

Контейнер зворотного управління в Laravel - потужний інструмент для управління залежностями класів. Впровадження залежностей є спосіб зняття жорстко закодованих залежностей класу. Замість цього залежності впроваджуються під час виконання, забезпечуючи більшу гнучкість, так як реалізація залежностей може бути легко змінена.

Ви можете використовувати IoC-контейнер в вашому додатку, щоб зробити все впроваджується і легко тестується за допомогою об'єктів-заглушок. IoC-контейнер може витягувати класи без будь-якої настройки.

Запити форм

У Laravel є чудова можливість, яка називається запитами форм. Це перевірка запитів для ваших контролерів. Це не просто перевірка даних, а повна обробка запиту.

Eloquent

Eloquent ORM, включена в Laravel, - найбільш потужна реалізація шаблону ActiveRecord в PHP. Крім звичайних CRUD-операцій, в ній є м'яке видалення, області запитів, відносини, методи доступу і мутатори, мутатори дат, спостерігачі моделі і багато іншого.

Цикл версій

У Laravel дуже передбачуваний цикл версій. Через кожні шість місяців можна чекати чергової версії. Laravel виходить через один місяць після виходу Symfony.

Ви отримуєте або версію з новими функціями, або версію з абсолютно новим номером. Версії з виправленнями помилок виходять кожні кілька

місяців. Це добре, тому що фреймворк розробляється дуже швидко, і ви можете розраховувати на отримання поліпшень безпеки та інших поліпшень так швидко, як тільки можливо.

SSH завдання

У Laravel є простий спосіб для SSH-підключень до віддалених серверів і запуску команд, що дозволяє вам легко створювати завдання, які працюють на віддалених серверах.

Flysystem

Flysystem - чудовий пакет з Ліги видатних пакетів для управління файловими системами. Це файлова система, яка використовує коннектори, за допомогою якої ви легко можете взаємодіяти з local, awss3, dropbox, rackspace і іншими файловими системами в хмарі.

Якщо ви користувалися новітніми технологіями для фронтенда останні кілька років, то, ймовірно, ви використовували препроцесор для ваших файлів CSS і JavaScript. Але якщо немає, то Laravel про це подбає.

Вам потрібна максимально оптимізована онлайн-версія вашої програми. Це означає, що вам треба зменшувати і комбінувати ваші файли CSS і JavaScript.


Замість використання для цього будь-яких PHP-пакетів, Laravel пропонує пакет Elixir для виконавця завдань NodeJS Gulp. Тому ви можете використовувати всі ті переваги NodeJS і Gulp, які вам подобаються.

Ви запускаєте Gulp-завдання watch, і Elixir подбає про все для вас. Якщо ви використовуєте LESS і CoffeeScript:

```

1  /// <reference path='../_all.ts' />
2
3  module todos {
4      'use strict';
5
6      /**
7       * Services that persists and retrieves TODOs from localStorage.
8       */
9      export class TodoStorage implements ITodoStorage {
10
11
12
13
14
15      }
16
17

```



compiled.php

Будь повнофункціональний фреймворк завантажується з тисячею файлів і оголошень. Здебільшого, всі вони обробляються автозавантаженням Composer, але для найбільш використовуваних файлів Artisan може створити файл `compiled.php`, який завантажується при кожному запиті і об'єднує всі класи, які використовуються в вашому додатку.

Найбільш використовувані файли і класи компілюються в один-єдиний файл, який значно зменшує час завантаження програми. Якщо у вас є файли, які потрібно завантажувати при кожному запиті, ви також можете вказати, які файли компілювати в `compiled.php`.

Потім цей файл оптимізується виконуючою середовищем PHP, тому ви можете уникнути зниження продуктивності при завантаженні тисяч файлів в ваше робоче оточення.

HHVM

Коли ви захочете підвищити продуктивність вашого PHP-коду, ви можете змінити PHP-оточення на високооптимізовані середовища виконання для PHP від Facebook - HHVM.

Laravel - один з перших фреймворків, що підтримують HHVM, і він проходить 100% юніт-тестів на HHVM. Тому ви можете бути впевнені, що легко можете переключити ваше додаток з PHP на HHVM і отримати ще більше продуктивності.

Cashier

Laravel Cashier забезпечує виразний, гнучкий інтерфейс для послуг білінгових підписок Stripe. Він обробляє майже всі шаблонні коди білінгових підписок, які ви боїтеся написати.

На додаток до основного управління підписками, Cashier може обробляти купони, підміну підписок, «кількість» підписок, періоди знижок і навіть генерувати PDF-файли рахунків.

Спільнота і документація

Сьогодні кожен фреймворк має велике і дружнє співтовариство, і Laravel - не виняток. Люди, що підтримують і розробляють Laravel-пакети, дуже доброзичливі, і ви легко можете брати участь у створенні будь-якого пакета, і ваші правки будуть прийняті дуже скоро.

Це сприяє розширенню спільноти, і вам варто знати, що Laravel був внесений до списку найбільш популярних PHP-фреймворків в 2013 році, випередивши Phalcon, Symfony2 і CodeIgniter. А в серпні 2014 року Laravel став найпопулярнішим і переглядається PHP-проектом на GitHub.

Laravel є найпопулярнішим фреймворком на сьогоднішній день і продовжує стрімко розвиватися. Зараз Laravel - це величезна екосистема, що включає хостинг та платформу для розгортання додатків. Сьогодні найбільше проектів, що розробляються за допомогою фреймворків, створюються саме з використанням Laravel.

2.3. ВИСНОВОК ДО РОЗДІЛУ 2

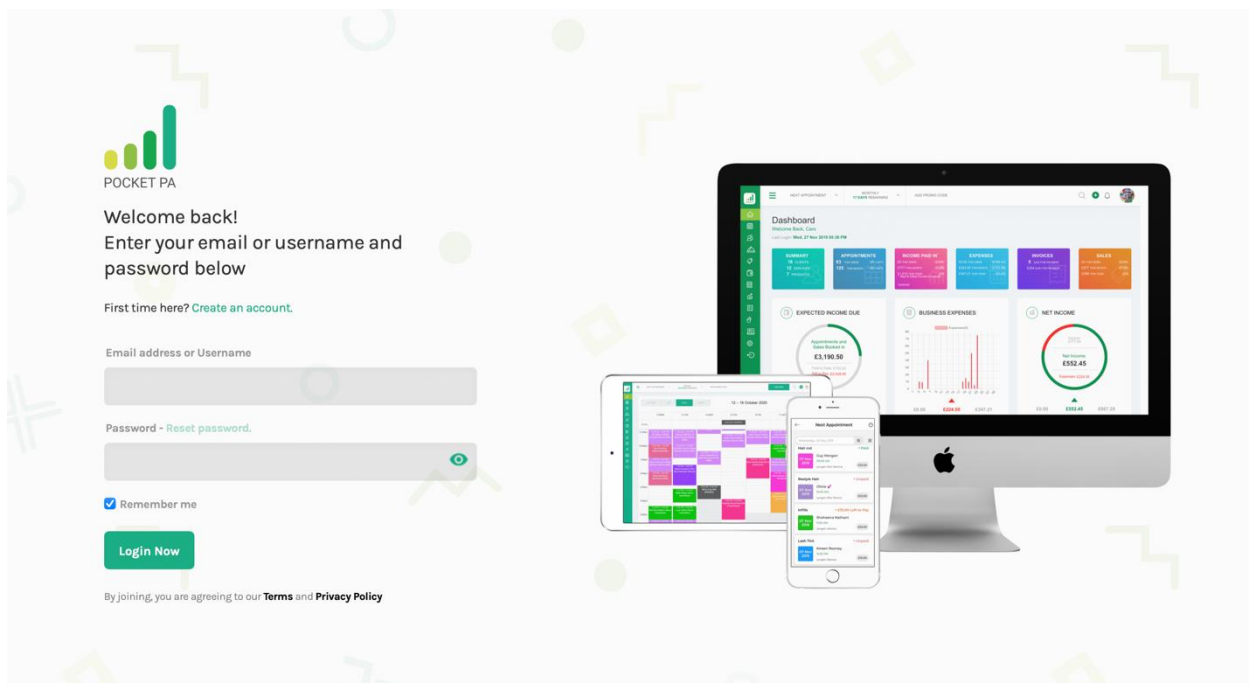
У цьому розділі проведено порівняльний аналіз PHP фреймворків, що використовуються при створенні різного роду веб додатків. Описано їх переваги та недоліки. Оптимальним рішенням обрано фреймворк Laravel, найпопулярніший PHP фреймворк сьогодення. Наведено його сильні сторони, надано опис його основних особливостей. В якості середовища розробки обрано IDE VSCode, оскільки це не просто редактор коду, а повноцінна безкоштовна середовище розробки з безліччю різноманітних та дуже корисних пакетів, які полегшують процес розробки.

РОЗДІЛ 3. РОЗРОБКА ВЕБ ДОДАТКА

3.1. Реалізація проекту

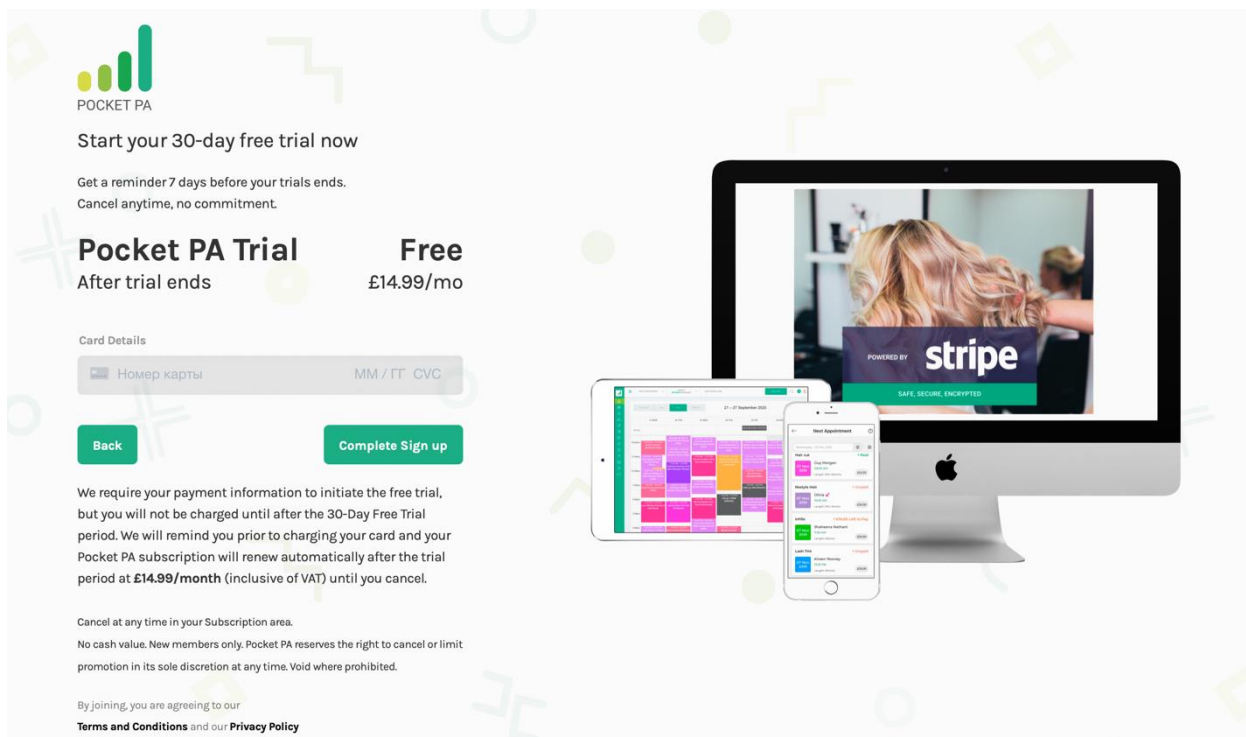
Як зазначено у висновку розділу 2 PHP фреймворк Laravel є найкращою базою для створення веб додатку обліку фінансів. Для роботи з даними застосовується система управління реляційними базами даних MySQL. Це система з відкритим кодом. Одна з найбільш використовуваних систем управління базами даних в програмних продуктах. Має хорошу підтримку з боку мови програмування PHP. Також для роботи з клієнтом у браузері використовується JS фреймворк VUEJS, який посідає перші місця по застосовності серед розробників SaaS платформ, на кшталт нашої.

На початку було необхідно розробити систему автентифікації та авторизації користувачів у систему.



Веб додаток є платним. Тому форма реєстрації повинна містити поле для заповнення банківської картки. Безпека передачі даних забезпечується

протоколом HTTPS та сервісом Stripe, задача якого проводити платежі та автоматичні списання коштів за користування веб додатком.



Фреймворк Laravel надає готові методи роботи з сервісом Stripe, за допомогою якого можливою автоматично створювати платежі для користувачів, взаємодіяти з банківськими рахунками та реальними коштами. Нижче надано приклад алгоритму роботи створення платежу за користування веб додатком.

```

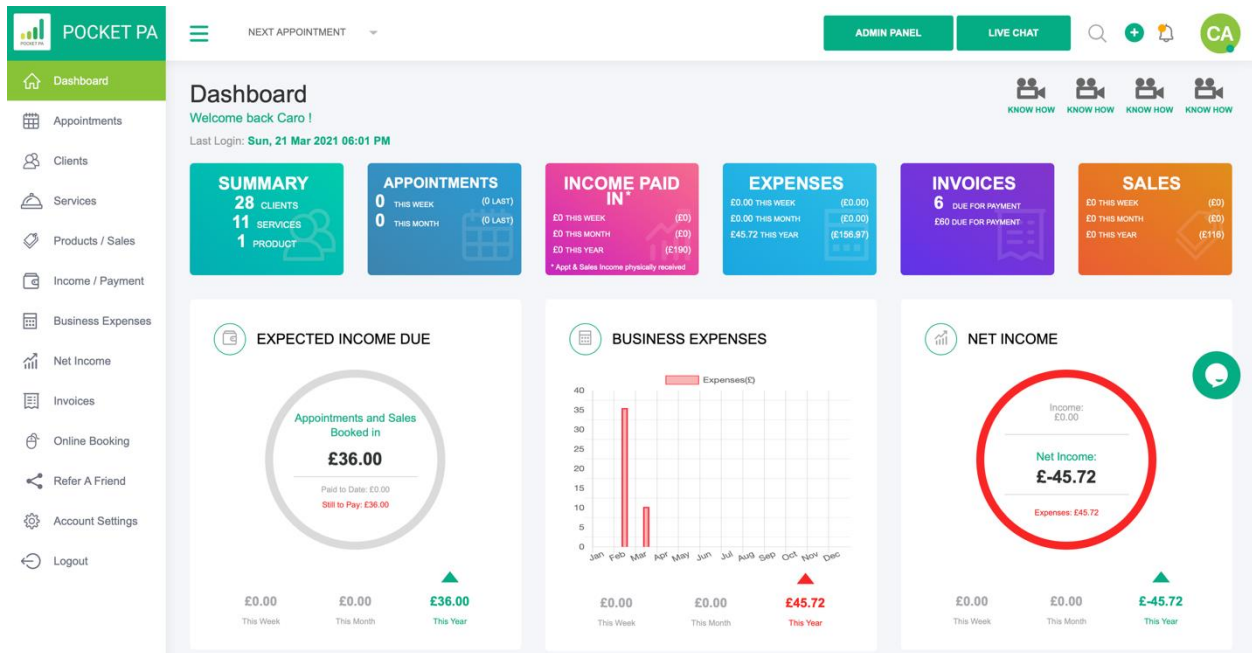
.../**
... * @param User $user
... * @param string $token
... * @return User
... */
... public function attachPaymentMethod(User $user, $token)
... {
...     $source = $this->client->customers->createSource(
...         $user->stripeId(),
...         ['source' => $token]
...     );
...     return $user->updateDefaultPaymentMethodFromStripe();
... }

.../**
... * @param User $user
... * @return \Laravel\Cashier\Subscription
... */
... public function createTrialSubscription(User $user)
... {
...     $monthlyPlan = collect($this->getPlans())
...         ->firstWhere('interval_count', 1);
...     $subscription = $user->newSubscription('default', $monthlyPlan->id);
...     if ($user->hasPromoCode()) $subscription->withCoupon($user->promo_code);
...     return $subscription->trialDays(30)->create(null, ['email' => $user->email]);
... }

.../**
... * @param User $user
... * @param $planId
... * @return \Laravel\Cashier\Subscription
... */
... public function createSubscription(User $user, $planId)
... {
...     $subscription = $user->newSubscription('default', $planId);
...     if ($user->hasPromoCode()) $subscription->withCoupon($user->promo_code);
...     // TODO: add newsletters (maybe through events where Slack notifications will be placed too)
...     return $subscription->create(null, ['email' => $user->email]);
... }

```

Важливою складовою SaaS платформи є інтуїтивний інтерфейс, тому було прийнято рішення створити інтерактивні макети за допомогою фреймворка VueJS. Першими було розроблено дизайн та інтерактивні частини головної сторінки додатку.



```

<ul class="at-statistics" :class="widthClass">
  <li v-if="dashboard_settings.dashboard_summary">
    <div class="at-statistic at-clients text-center">
      <h3><strong>SUMMARY</strong></h3>
      <h3><a href="/vendors/clients"><strong>{{clients}}</strong>{{clientsCase()}}</a></h3>
      <h3><a href="/vendors/services"><strong>{{services}}</strong>{{servicesCase()}}</a></h3>
      <h3><a href="/vendors/products"><strong>{{products}}</strong>{{productsCase()}}</a></h3>
      <i class="icon-Clients"></i>
    </div>
  </li>
  <li v-if="dashboard_settings.dashboard_appointments">
    <div class="at-statistic at-appointment">
      <a href="/vendors/appointments">
        <h3 class="text-center"><strong class="at-stats-appt">APPOINTMENTS</strong></h3>
        <h3><strong>{{appointments_of_week}}</strong> <small>this week</small> <span class="right">{{appointments_of_
        <h3><strong>{{appointments_of_month}}</strong> <small>this month</small> <span class="right">{{appointments_of_
        <i class="icon-Calendar"></i>
      </a>
    </div>
  </li>
  <li v-if="dashboard_settings.dashboard_income">
    <div class="at-statistic at-services">
      <a href="/vendors/payments">
        <h3 class="text-center"><strong>INCOME PAID IN<small style="vertical-align: top;padding-left: 2px;line-height:
        <h3>
          <formatted-amount :after-dot="0" :currency="currency" :amount="parseFloat(stats.thisWeekIncome)"></format
          <small>this week</small> <span class="right">(<formatted-amount :after-dot="0" :currency="currency" :amou
        <h3>
          <formatted-amount :after-dot="0" :currency="currency" :amount="parseFloat(stats.thisMonthIncome)"></format
          <small>this month</small> <span class="right">(<formatted-amount :after-dot="0" :currency="currency" :amou
        <h3>
          <formatted-amount :after-dot="0" :currency="currency" :amount="parseFloat(stats.thisYearIncome)"></format
          <small>this year</small> <span class="right">(<formatted-amount :after-dot="0" :currency="currency" :amou
        <i class="icon-net-income"></i>
      </a>
    </div>
    <small class="at-statistic-small">* Appt & Sales Income physically received</small>
  </li>
</ul>

```

Задля ефективного обліку фінансів веб додаток потрібен містити та обробляти продукти на сервіси, які користувач може продавати та надавати клієнтам для подальшого отримання коштів та їх підрахунків. Тому були розроблені відповідні сторінки.

POCKET PA NEXT APPOINTMENT

ADMIN PANEL LIVE CHAT

Services

Search by service name, price

SERVICE NAME	DURATION	PRICE	PRODUCTIVITY / HOURLY RATE
Haircut	01:00 Hr / Mins	£25.00	£25.00 / Hour
Wax	00:15 Hr / Mins	£6.00	£24.00 / Hour
Highlights	01:30 Hr / Mins	£50.00	£33.33 / Hour
Lash Tint	00:10 Hr / Mins	£7.00	£42.00 / Hour
Trim Hair Cut	00:45 Hr / Mins	£15.00	£20.00 / Hour
TEST	00:15 Hr / Mins	£15.00	£60.00 / Hour
TEST	00:15 Hr / Mins	£15.00	£60.00 / Hour

POCKET PA NEXT APPOINTMENT

ADMIN PANEL LIVE CHAT

Sales

WEEK MONTH YEAR < 2020 >

1-Jan-2020 31-Dec-2020 Search Sales

DATE	PRODUCT NAME	CLIENT	QTY	PAYMENT METHOD	STATUS	UNIT PRICE	TOTAL PRICE
Tue, 04 Feb 2020	Green Hat	Meg	1	CASH	• £5 LEFT TO PAY	£12.00	£12.00
Sun, 16 Feb 2020	Green Hat	Caroline	1	INVOICE	• UNPAID	£12.00	£12.00
Sun, 16 Feb 2020	Green Hat	Caroline	1	INVOICE	• UNPAID	£12.00	£12.00
Mon, 17 Feb 2020	Green Hat	Caroline	1	INVOICE	• UNPAID	£12.00	£12.00
Mon, 17 Feb 2020	Green Hat	Caroline	1	INVOICE	• UNPAID	£12.00	£12.00

Після створення сервісів та продуктів користувач має додати платежі, які отримав від клієнтів. Система автоматично підрачує, скільки коштів надійшло у який час, яка частина коштів ще очікується та список надходжень. Нижче наведені дизайн сторінки платежів та приклад алгоритму підрахунку надходжень за наданий період.

POCKET PA

☰
NEXT APPOINTMENT

ADMIN PANEL
LIVE CHAT
🔍
+
📧
CA

- [Dashboard](#)
- [Appointments](#)
- [Clients](#)
- [Services](#)
- [Products / Sales](#)
- [Income / Payment](#)
- [Business Expenses](#)
- [Net Income](#)
- [Invoices](#)
- [Online Booking](#)
- [Refer A Friend](#)
- [Account Settings](#)
- [Logout](#)

Appointments Payments
Sales Payments

🔍
20
▼

DATE	CLIENT	AMOUNT	SERVICE	PAYMENT	STATUS	
Fri, 17 Jan 2020	Sally	£1.00	Wax	CARD PAYMENT	● Paid	
Fri, 17 Jan 2020	Sally	- £1.00	Wax	CARD PAYMENT	● Refund	
Mon, 27 Jan 2020	Meg	£10.00	Haircut	CASH	● Paid	
Tue, 04 Feb 2020	Alison Blair	£25.00	Haircut	CASH	● Paid	
Tue, 04 Feb 2020	Alison Blair	£1.00	Wax	CASH	● Paid	
Tue, 04 Feb 2020	Alison Blair	£4.00	Wax	CASH	● Paid	
Sat, 15 Feb 2020	Meg	£50.00	Highlights	CASH	● Paid	
Mon, 10 Aug 2020	Alison Blair	£15.00	Highlights	CASH	● Paid	
Mon, 07 Sep 2020	Caro TESTING	£56.00	Highlights, Wax	CASH	● Paid	
Total		£161.00				

SHOWING 1 TO 9 OF 9 APPOINTMENTS PAYMENTS

POCKET PA

☰
NEXT APPOINTMENT

ADMIN PANEL
LIVE CHAT
🔍
+
📧
CA

- [Dashboard](#)
- [Appointments](#)
- [Clients](#)
- [Services](#)
- [Products / Sales](#)
- [Income / Payment](#)
- [Business Expenses](#)
- [Net Income](#)
- [Invoices](#)
- [Online Booking](#)
- [Refer A Friend](#)
- [Account Settings](#)
- [Logout](#)

WEEK
MONTH
YEAR

<
2020
>

1-Jan-2020
31-Dec-2020

EXPECTED INCOME DUE

Appointments Booked In:	£908.00
Sales Made:	£116.40
Paid to Date:	£189.60
Still to Pay:	£834.80

£1,024.40

Appointments and Sales Booked In

Paid to Date: £189.60

Still to Pay: £834.80

PHYSICALLY RECEIVED PAYMENTS AND REFUNDS

Show Income Details

Payments Due In 2020

Monies carried forwards in 2020	£0.00
Monies from appointments in 2020	£161.00
Monies from sales in 2020	£28.60
Monies received after 2020	£0.00

Still To Pay

Upcoming appointments during 2020	£0.00
Appts booked in 2020 (Marked as 'pending' but not yet paid for)	£747.00
Appts booked in 2020 (marked as 'attended') but not yet paid for	£0.00
Sales made in 2020 but not yet paid for	£87.80

Cancelled Appointments Payments

Money received for cancelled appointments	£0.00
Total	£0.00

Monies Taken For Future / Historic Periods

Monies paid in 2020 but for sales & appointments before 2020	£0.00
Monies paid in 2020 but for work/sales after 2020	£0.00


```

$startDateToDateString = Carbon::parse($request->startDate)->toDateString();
$endDateToDateString = Carbon::parse($request->endDate)->toDateString();

if(authUser()->isVendor()){

    /*paid to date*/
    $apsPaidToDate = authUser()->appointments()
        ->whereDate('date_time','>=',$startDateToDateString)
        ->whereDate('date_time','<=',$endDateToDateString)
        ->where('status','!=','cancelled')
        ->whereHas('payments', function ($q) use($startDateToDateString,$endDateToDateString) {
            $q->whereDate('date','>=',$startDateToDateString);
            $q->whereDate('date','<=',$endDateToDateString);
        })
        ->get();

    $amountsPaidToDate= 0;
    $refundsPaidToDate= 0;

    foreach ($apsPaidToDate as $ap){
        $amountsPaidToDate+=$ap->payments()
            ->whereDate('date', '>=',$startDateToDateString)
            ->whereDate('date', '<=',$endDateToDateString)
            ->where('is_refund',0)->sum('amount');
        $refundsPaidToDate+=$ap->payments()
            ->whereDate('date', '>=',$startDateToDateString)
            ->whereDate('date', '<=',$endDateToDateString)
            ->where('is_refund',1)->sum('amount');
    }
    $payments['appointments_paid_to_date'] = ($amountsPaidToDate-$refundsPaidToDate);
    /*paid to date*/

    /*still to pay*/
    $payments['appointments_still_to_pay'] = authUser()->appointments()
        ->whereDate('date_time','>=',$startDateToDateString)
        ->whereDate('date_time','<=',$endDateToDateString)
        ->where('status','!=','cancelled')
        ->get()->sum('pending');
    /*still to pay*/

```

Алгоритм підрахунку доволі важкий з точки зору витрати ресурсів сервера, тому було прийнято рішення перенести більшість процесів обрахунку до системи MYSQL.

Для максимальної автоматизації надходжень коштів було також застосовано сервіс Stripe – а саме систему інвойсів – автоматичних електронних email-ів з можливістю оплати картою. Користувач створює такий інвойс, заповнює його наданими сервісами або продуктами та відправляє по електронній пошті. Клієнт отримує сторінку, де зможе заповнити дані своєї банківської карти та оплатити надані послуги. Веб додаток автоматично збереже та підрахує отриманий прибуток.

Add Invoice

Client **Email**

Aaron Benchmarks michael.faulkner@benchmarjoinery.co.uk

Date	Product(s)	Quantity	Amount
22-Jun-2021	Green Hat	2	£ 24.00

Have you already received any part payment on this?

+ ADD NEW PRODUCT
+ ADD NEW APPOINTMENT

Total Amount	£24.00
Payments Made	£0.00
Left To Pay	£24.00
Invoice Amount £24.00	

Notes

Type in any message here that you wish to send to the client you are invoicing

CANCEL
PREVIEW
SAVE & SEND

ZOMSCO Ltd

£14.99

To: blakevcaryl1@gmail.com
From: ZOMSCO Ltd

View invoice details >

Or pay with card

Card information

1234 1234 1234 1234 VISA M/C

MM / YY CVC

Pay £14.99

Powered by stripe | Terms Privacy

Окрім ведення обліку фінансів користувачі мають змогу автоматизувати процес надання послуг за допомогою сторінки календаря та відміток надання послуг для бази клієнтів. Веб додаток надає змогу зберігати та сортувати базу клієнтів для кожного користувача системою.

POCKET PA NEXT APPOINTMENT ADMIN PANEL LIVE CHAT

Search by name, e-mail or mobile number 20 EXPORT

CLIENT NAME	EMAIL	PHONE	APPOINTMENTS	PAYMENTS	SALES	INVOICES	LAST APPOINTMENT DATE
KR Kayleigh Rotherham	kay.rotherham@hotmail.co.uk	+447341847478	0	0	0	0	-
LV Landlord Vaion			0	0	0	0	-
Meg	megansyson@hotmail.com	+447972847146	5	3	1	0	Sun, 02 Feb 2020
Olivia	boolovesmanu@hotmail.co.uk	+447786431546	0	0	0	0	-
SA Sally	carolinesyson@gmail.com	+447702664393	7	2	0	0	Tue, 28 Jan 2020
SA Sam		+447454015179	0	0	0	0	-
SR Sam Robertson		+447872676167	0	0	0	0	-
Zak		+447725990542	0	0	0	0	-

SHOWING 21 TO 28 OF 28 CLIENTS

POCKET PA NEXT APPOINTMENT ADMIN PANEL LIVE CHAT

EXPORT

27 January – 2 February 2020

	27 MON	28 TUE	29 WED	30 THU	31 FRI	1 SAT	2 SUN
all-day							
11:00am							
12:00pm					12:15 PM - 1:15 PM Alison Blair (Haircut)	12:00 PM - Alison Blair (Wax)	
1:00pm							1:10 PM - Alison Blair (Wax)
2:00pm		1:50 PM - 3:20 PM Meg (Highlights)	1:50 PM - 3:20 PM Meg (Highlights)			1:50 PM - 3:20 PM Meg (Highlights)	1:50 PM - 3:20 PM Meg (Highlights)
3:00pm							
4:00pm		3:30 PM - 4:30 PM Sally (Haircut)					4:00 PM - 5:00 PM Abbey Jacobs (Haircut)
5:00pm							

Додаток надає змогу автоматично посилати повідомлення для клієнтів через систему CRONJOB у PHP. Був розроблений відповідний алгоритм та інтерфейс взаємодії з такими повідомлення у веб додатку. Нижче наведені приклади інтерфейсу та алгоритму.

POCKET PA ADMIN PANEL LIVE CHAT

Client Notification Area / Email Notification

Enable Notification Email
 Activate automatic emails to your clients to remind them of their upcoming appointment or to thank them after the appointment.
 Emails are sent from no_reply@pocketpa.com so clients will not be able to respond directly to you

Booking Confirmation | Reminder Email | Thank You Message

Enable Booking Confirmation Email

Email Subject
 Appointment Booking Confirmation for your SERVICE_NAME from BUSINESS_NAME

Email Template
 Hi CLIENT_FIRST_NAME,
 This is your appointment booking confirmation for your SERVICE_NAME
 Here are the details:
 APPOINTMENT_DATE_TIME
 At LOCATION_NAME
 LOCATION_PHONE
 We look forward to seeing you soon
 Best wishes
 RUSINFSS_NAME

Email Settings
 When you use the below tags in your template, be very careful to copy and paste the words correctly as any missing/extra letters or underscore (_) will affect the message and details which get populated and sent.
 You can also add a couple of emojis to really make it pop!

Use the below tags to include appointment details inside messages:

- CLIENT_FIRST_NAME
- CLIENT_LAST_NAME
- APPOINTMENT_DATE_TIME
- APPOINTMENT_DATE
- APPOINTMENT_TIME
- SERVICE_NAME
- BUSINESS_NAME
- LOCATION_NAME
- LOCATION_PHONE

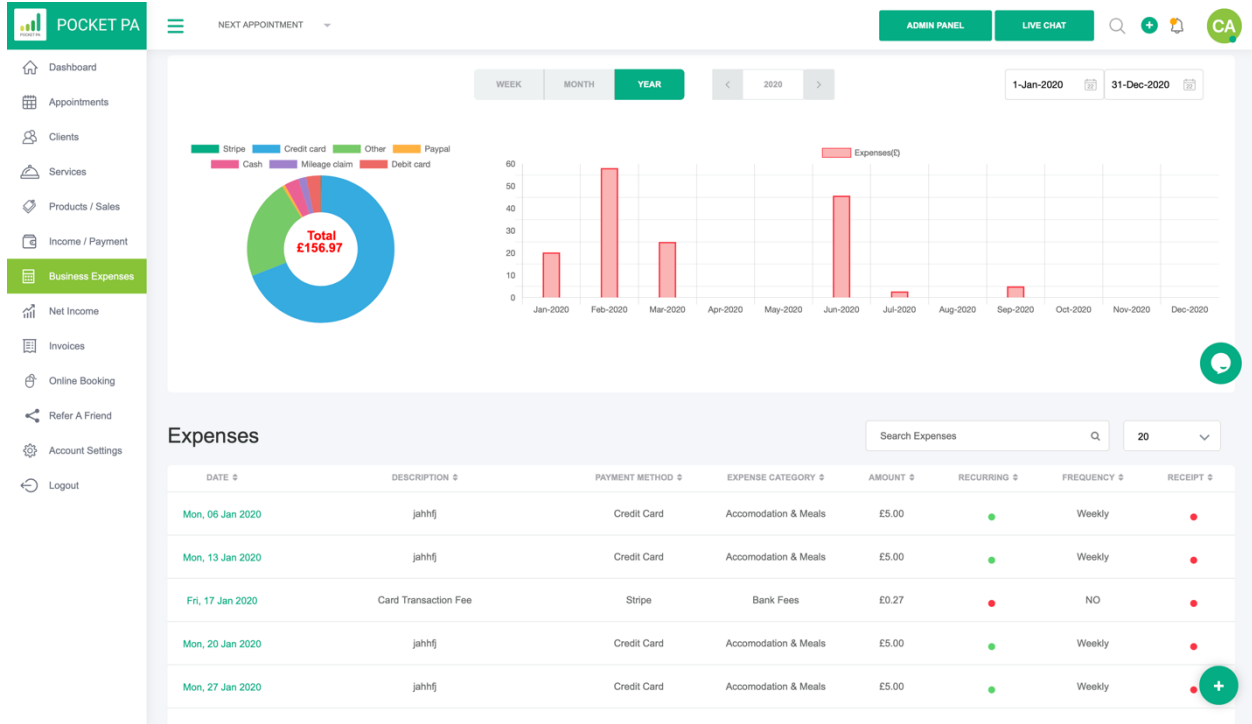
SAVE CHANGES

```
//get email settings
if ($setting->is_email_notification_enabled && $app->client->is_email_reminder_active) {
    if ($setting->is_email_reminder_enabled
        && $diffInMinutes <= ($setting->send_email_reminder_before * 60)
        && !$isPast
        && !$notification->is_notification_sent) {
        if ($app->client->email) {
            $rows[] = [
                'app id' => $app->id,
                'user' => $vendor->fullname(),
                'client' => $client->name,
                'service_name' => $app->service_name ?? 'N/A',
                'date_time' => $app->unformatted_date_time,
                'send notification in' => $diffInMinutes . ' minutes before appt',
                'reminder type' => 'email',
            ];

            SendNotificationEmail::dispatch($notification->id, [
                'field_to_update' => 'is_notification_sent',
                'appointment_id' => $app->id,
                'subject' => 'email_reminder_subject',
                'template' => 'reminder_email_template',
                'vendor_id' => $vendor->id,
                'type' => Message::TYPE_REMINDER
            ]);
        } else {
            $this->info('Appt Notification: client ' . $app->client->id . ' ' . $app->client->full_name . ' does not have email
        }
    }
}
```

Відправка великої кількості повідомлень може погано відобразитися на роботі сервера, тому було прийнято рішення винести відправку повідомлень в спеціальний сервіс AWS SQS, який спрямований на роботу з відкладеною логікою, як у системі CRONJOB.

Облік фінансів неможливий без урахування витрат. Тому було поставлено задачу розробити сторінку для внесення витрат по бізнесу з можливістю пролонгованих та періодичних витрат, які формуються автоматично.



Business Receipt Form

Amount: £ 0.00 | Date: 22-Jun-2021

Recurring:

Expense description:

Expense Categories:

Payment Method Selection:

- CASH
- BANK PAYMENT
- CREDIT CARD
- DEBIT CARD
- PAYPAL
- CHEQUE
- GIFT VOUCHER
- OTHER

Buttons: CANCEL, SAVE

Нижче наведено частину алгоритму створення періодичних витрат по декільком параметрам.

```

public function handle()
{
    //find recurring expenses
    //save into new array
    //loop over them and create new if recurring interval comes today using start_date and interval
    $expensesCronJobs = RecurringExpenseCronJob::where('next_recurrence', '<=', today())->with('expense')->get();

    if(count($expensesCronJobs) > 0){
        $rows = [];
        foreach ($expensesCronJobs as $expenseJob){
            $nextRecurrence = $expenseJob->next_recurrence;

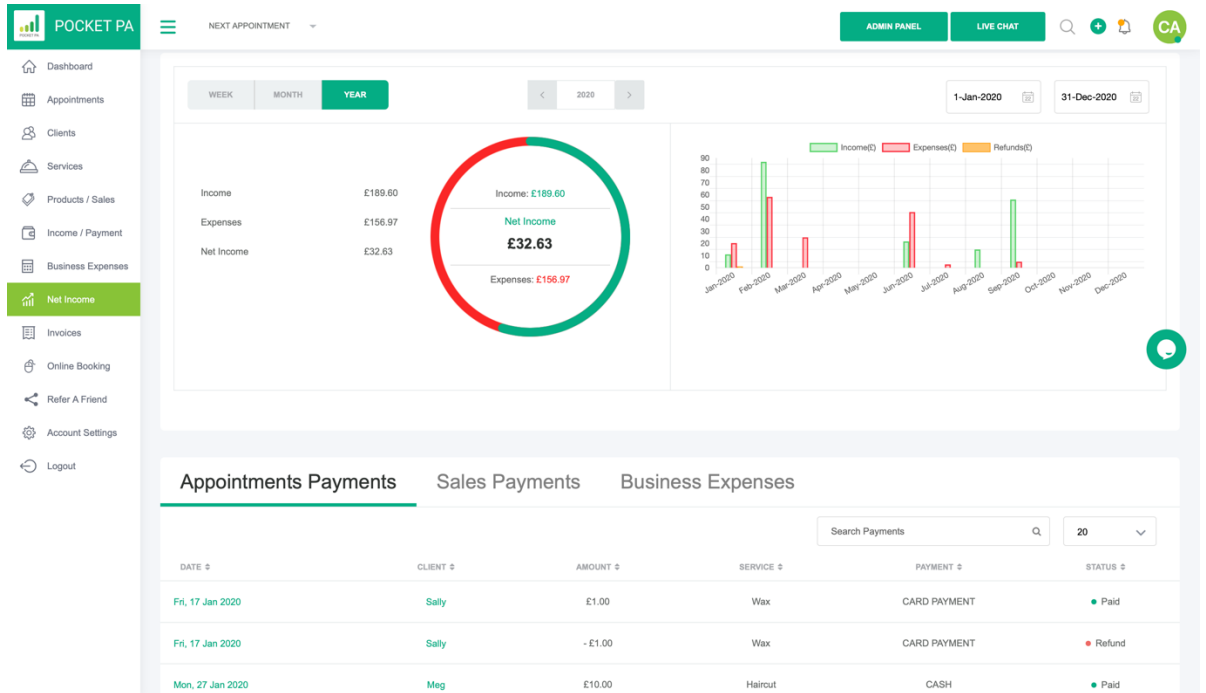
            while (Carbon::parse($nextRecurrence)->lte(today()) {
                $rows[] = [
                    'expense_id' => $expenseJob->expense->id,
                    'amount' => $expenseJob->expense->amount,
                    'start_date' => $expenseJob->start_date,
                    'end_date' => $expenseJob->recurring_end_date,
                    'frequency' => $expenseJob->frequency,
                    'total' => $expenseJob->total_recurrence,
                    'remaining' => $expenseJob->remaining_recurrence,
                ];
                $data = [];
                $data['parent_expense_id'] = $expenseJob->expense->id;
                $data['user_id'] = $expenseJob->expense->user_id;
                $data['start_date'] = $nextRecurrence;
                $data['description'] = $expenseJob->expense->description;
                $data['payment_method'] = $expenseJob->expense->unformatted_payment_method;
                $data['expense_category'] = $expenseJob->expense->expense_category;
                $data['amount'] = $expenseJob->expense->unformatted_amount;
                $data['recurring'] = 1;
                $data['recurring_end_date'] = $expenseJob->expense->unformatted_recurring_end_date;
                $data['receipt_image'] = $expenseJob->expense->receipt_image;
                $data['frequency'] = $expenseJob->expense->unformatted_frequency;

                Expense::create($data);

                $expenseJob->update(['next_recurrence' => nextRecurrence($expenseJob)]);
                $nextRecurrence = $expenseJob->next_recurrence;
            }
        }
        $this->table(['expense id','amount','start date','end date','frequency','total','remaining'],$rows);
    }else{
        $this->info('No recurring expenses of today found...');
    }
}

```

Користувач має можливість оцінити прибутковність свого бізнесу. Тому була розроблена сторінка зведених даних по прибутку та витратам та відповідний алгоритм для опрацювання та підрахунку чистого прибутку за винятком витрат. Нижче наведені приклади.



```

public function searchForBarChart($request,$query, $type = null)
{
    $differenceInDays = Carbon::parse($request->startDate)->diffInDays(Carbon::parse($request->endDate));
    if($differenceInDays == 0){
        $xAxisValue = 'day';
        if ($type != 'sale') {
            $mySqlFnName = 'DATE_FORMAT(payments.date,"%d-%b-%Y") as ';
        } else {
            $mySqlFnName = 'DATE_FORMAT(sales.date,"%d-%b-%Y") as ';
        }
    }
    else if($differenceInDays > 0 && $differenceInDays < 8){
        $xAxisValue = 'dayname';
        if ($type != 'sale') {
            $mySqlFnName = 'DAYNAME(payments.date) as ';
        } else {
            $mySqlFnName = 'DAYNAME(sales.date) as ';
        }
    }
    else if($differenceInDays >= 8 && $differenceInDays < 32){
        $xAxisValue = 'day';
        if ($type != 'sale') {
            $mySqlFnName = 'DATE_FORMAT(payments.date,"%d-%b-%Y") as ';
        } else {
            $mySqlFnName = 'DATE_FORMAT(sales.date,"%d-%b-%Y") as ';
        }
    }
    else if($differenceInDays > 32){
        $xAxisValue = 'month';
        if ($type != 'sale') {
            $mySqlFnName = 'DATE_FORMAT(payments.date,"%b-%Y") as ';
        } else {
            $mySqlFnName = 'DATE_FORMAT(sales.date,"%b-%Y") as ';
        }
    }
    if($request->filled('startDate') && $type != 'sale'){
        $query = $query->whereDate('payments.date','>=',Carbon::parse($request->startDate)->toDateString());
    } else {
        $query = $query->whereHas('sale', function($q) use ($request) {
            $q->whereDate('date','>=',Carbon::parse($request->startDate)->toDateString());
        });
    }
    if($request->filled('endDate') && $type != 'sale'){
        $query = $query->whereDate('payments.date','<=',Carbon::parse($request->endDate)->toDateString());
    }
}

```


Будь-який веб додаток має містити шляхи комунікації з клієнтом – браузером. Тому була розроблена система routes – спеціальних шляхів передачі даних по протоколу HTTPS. Нижче наведені приклади таких ліній передач.

```

/*PRODUCTS*/
Route::get('/products/export', 'ProductsController@export')->name('products.export');
Route::post('/products/search-by-ajax', 'ProductsController@searchByAjax')->name('products.search_by_ajax');
Route::get('/products/{id}/check', 'ProductsController@checkProduct')->name('products.check');
Route::resource('products', 'ProductsController');

/*SALES*/
Route::get('/sales/export', 'SalesController@export')->name('sales.export');
Route::get('/sales/{id}/check', 'SalesController@checkSaleToDelete')->name('sales.check');
Route::post('/sales/search-by-ajax', 'SalesController@searchByAjax')->name('sales.search_by_ajax');
Route::resource('sales', 'SalesController')->except(['index', 'show']);

/*SALES INVOICE*/
Route::get('/invoices/export', 'InvoicesController@export')->name('invoices.export');
Route::get('/invoice-preview/export', 'InvoicesController@exportPreview')->name('invoice_preview.export');

Route::post('/invoices/search-by-ajax', 'InvoicesController@searchByAjax')->name('invoices.search_by_ajax');

Route::post('/invoices/send', 'InvoicesController@send')->name('invoices.send');
Route::post('/invoices/{id}/remind', 'InvoicesController@remind')->name('invoices.remind');
Route::post('/invoices/{id}/void', 'InvoicesController@voidInvoice')->name('invoices.void');

Route::resource('invoices', 'InvoicesController')->except(['show', 'edit', 'store', 'update', 'destroy']);

/*MESSAGES */
Route::post('/messages/search-by-ajax', 'MessagesController@searchByAjax')->name('messages.search_by_ajax');

/*ACCOUNT-SETUP*/
Route::get('/user', 'AccountSetupController@getUser')->name('get_user');

Route::get('account-setup', 'AccountSetupController@main')->name('account_setup');
Route::post('account-setup/profile', 'AccountSetupController@updateProfile')->name('update_profile');
Route::post('account-setup/calendar-and-time-settings', 'AccountSetupController@calendarAndTimeSettings')->name('account_setup.calendar_and_time_settings');
Route::post('account-setup/payment-method-and-currency-settings', 'AccountSetupController@paymentMethodAndCurrencySettings')->name('account_setup.payment_method_and_currency_settings');
Route::post('account-setup/dashboard-settings', 'AccountSetupController@dashboardSettings')->name('account_setup.dashboard_settings');
Route::post('account-setup/export-settings', 'AccountSetupController@exportSettings')->name('account_setup.export_settings');

/*Locations*/
Route::get('account-setup/locations', 'AccountSetupController@getVendorAddressesByAjax')->name('account_setup.locations');
Route::get('account-setup/locations/{id}/check', 'AccountSetupController@checkVendorAddressToDelete')->name('account_setup.check_vendor_address_to_delete');
Route::get('account-setup/locations/{id}/check-online', 'AccountSetupController@checkVendorAddressOnlineBooking')->name('account_setup.check_vendor_address_online_booking');
Route::post('account-setup/locations', 'AccountSetupController@createAddress')->name('account_setup.create_address');
Route::get('account-setup/locations/{id}/edit', 'AccountSetupController@editAddress')->name('account_setup.edit_address');
Route::put('account-setup/locations/{id}', 'AccountSetupController@updateAddress')->name('account_setup.update_address');
Route::delete('account-setup/locations/{id}', 'AccountSetupController@deleteAddress')->name('account_setup.delete_address');

/* Affiliates */
Route::get('affiliates', 'AffiliateController@main')->name('affiliate.main');
Route::get('affiliates/signup', 'AffiliateController@signup')->name('affiliate.signup');
Route::get('affiliates/export', 'AffiliateController@export')->name('affiliates.export');

Route::post('affiliates/search-by-ajax', 'AffiliateController@searchByAjax')->name('affiliate.search_by_ajax');
Route::post('affiliates/payments/search-by-ajax', 'AffiliateController@paymentsSearchByAjax')->name('affiliate.payments.search_by_ajax');
Route::post('affiliates/search-by-ajax-for-pie-chart', 'AffiliateController@earningsForPieChartByDateRange')->name('affiliate.search_by_ajax_for_pie_chart');
Route::post('affiliates/search-by-ajax-for-bar-chart', 'AffiliateController@affiliatesForBarChartByDateRange')->name('affiliate.search_by_ajax_for_bar_chart');

/*ONLINE BOOKING*/
Route::get('online-booking/{slug}', 'OnlineBookingsController@index')->name('online_booking.index');
Route::get('online-booking/{slug}/get-services', 'OnlineBookingsController@services')->name('online_booking.services');
Route::post('online-booking/{slug}/get-location', 'OnlineBookingsController@getLocation')->name('online_booking.location');
Route::post('online-booking/{slug}/get-available-hours', 'OnlineBookingsController@getAvailableHours')->name('online_booking.get_available_hours');
Route::post('online-booking/{slug}/verify-email-address', 'OnlineBookingsController@verifyEmailAddress')->name('online_booking.verify_email_address');
Route::post('online-booking/{slug}/verify-confirmation-code', 'OnlineBookingsController@verifyConfirmationCode')->name('online_booking.verify_confirmation_code');
Route::post('online-booking/{slug}/get-client-and-service-info', 'OnlineBookingsController@getClientAndServiceInfo')->name('online_booking.get_client_and_service_info');
Route::post('online-booking/{slug}/book-appointment', 'OnlineBookingsController@bookAppointment')->name('online_booking.book_appointment');

```

```

Route::get('/welcome','VendorsController@landing')->name('welcome');
Route::get('/onboard','VendorsController@onboard')->name('onboard');
Route::get('/search-data','VendorsController@search');

Route::get('/get-addresses','VendorsController@getAddressesByAjax')->name('addresses_by_ajax');

Route::get('/mileage-amount','VendorsController@getMileageAmount')->name('mileage_amount');
Route::post('/count-miles','VendorsController@countMiles')->name('count_miles');

Route::post('/delete-address','VendorsController@deleteAddressByAjax')->name('delete_address_by_ajax');
Route::post('/delete-phone','VendorsController@deletePhoneByAjax')->name('delete_phone_by_ajax');

Route::put('/notifications/{id}/mark-as-read','VendorsController@readNotification')->name('notification.mark_as_read');
Route::put('/notifications/mark-all-as-read','VendorsController@readAllNotifications')->name('notification.mark_all_as_read');
Route::put('/notifications/{id}/delete','VendorsController@deleteNotification')->name('notification.delete');

Route::put('/news/{id}/mark-as-read','VendorsController@readNews')->name('news.mark_as_read');
Route::put('/news/mark-all-as-read','VendorsController@readAllNews')->name('news.mark_all_as_read');
Route::put('/news/{id}/delete','VendorsController@deleteNews')->name('news.delete');

Route::post('/tutorial/seen','VendorsController@seenTutorialStep')->name('tutorial.seen');

/*SUBSCRIPTION PLANS*/
Route::namespace('Vendor')->group(function() {
    /*DASHBOARD*/
    Route::get('/dashboard','DashboardController')->name('dashboard');

    /*SUBSCRIPTION*/
    Route::get('/subscriptions','SubscriptionsController@index')->name('subscription.index');

    Route::get('/subscriptions/charge','SubscriptionsController@chargeSubscription')->name('charge.subscription');
    Route::get('/subscriptions/change/auto-renewal','SubscriptionsController@changeAutoRenewalSubscription')->name('autorenewal.subscription');
    Route::get('/subscriptions/cancel','SubscriptionsController@cancelSubscription')->name('cancel.subscription');

    Route::get('/subscriptions/set-default-card/{card_id}','SubscriptionsController@setDefaultCard')->name('set_default_card');
    Route::post('/subscriptions/add-card','SubscriptionsController@addCard')->name('add_card');
    Route::post('/subscriptions/remove-card/{card_id}','SubscriptionsController@removeCard')->name('remove_card');

    Route::get('/subscriptions/add-promo-code','SubscriptionsController@addPromoCode')->name('add.promo_code');
    Route::post('/subscriptions/find-promo-code','SubscriptionsController@findPromoCode')->name('find.promo_code');
    Route::post('/subscriptions/proceed-to-checkout','SubscriptionsController@proceedToCheckout')->name('proceed_to_checkout');
    Route::post('/subscriptions/make-subscription','SubscriptionsController@stripeCheckout')->name('make.subscription');
    Route::post('/subscriptions/switch-plan','SubscriptionsController@switchPlan')->name('switch.plan');
});

```

Кожен запит до сервера повинен містити один із заданих шляхів. Алгоритм автоматично розпізнає цей шлях та виконає необхідну логіку, яка закріплена за кожним шляхом, як видно на наведених прикладах. Така виконувана логіка називається контролерами, які є звичайними РНР класами зі вбудованими алгоритмами опрацювання запиту та надання відповіді до клієнта – браузера. Нижче наведено приклад такого контролера.


```

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    try {
        $response = $this->payment->find($id);
        $response = $this->paymentServiceInterface->find($id);
        if(array_key_exists('payment',$response)){
            $payment = $response['payment'];
            return view($this->folder.'.show',compact('payment'));
        }
        return redirect()->route($this->route.'.index')->with($response['status'],$response['message']);
    } catch (\Exception $e) {
        return $this->vendorDashboard($e,'Payment could not be found','vendor.payments.index');
    }
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    try {
        $response = $this->payment->find($id);
        if(array_key_exists('payment',$response)){
            $payment = $response['payment'];
            $previousUrl = url()->previous();
            return view($this->folder.'.edit',compact('payment','previousUrl'));
        }
        return redirect()->route($this->route.'.index')->with($response['status'],$response['message']);
    } catch (\Exception $e) {
        return $this->vendorDashboard($e,'Payment could not be found','vendor.payments.index');
    }
}

```

Контролер має опрацювати вхідні дані та направити їх до відповідного сервісу – іншого PHP класу зі своєю логікою роботи. Така система роботи з алгоритмами називається MVC – Model View Controller, де головна задача – це прозорий, зрозумілий та структурований код.

3.2. ВИСНОВОК ДО РОЗДІЛУ 3

В цьому розділі було описані основні методи та підходи для реалізації інтерактивного веб-сервісу для обліку фінансів. Також було реалізовано функціонал системи відповідно до вимог та наведено фрагменти програмного коду з описом їх роботи. Наведено приклади алгоритмів підрахунку та результати їх виконання.

ВИСНОВКИ

Під час виконання дипломного проекту було досліджено та випробувано методи створення веб додатків. Проведено порівняльний аналіз ефективності використання різних способів для розробки та тестування готового веб додатку. Було порівняно PHP фреймворки, наведено їх основні переваги та недоліки. Для реалізації проекту було обрано фреймворк Laravel, оскільки він є найпоширенішим та має безліч переваг над іншими.

В результаті дипломного проекту було отримано робочу saas платформу для обліку фінансів, яка налічує зараз більше 300 зареєстрованих користувачів. Побудований весь сервіс постійно розвивається та розширюється. Додаються нові функціональні можливості, такі як, календар зустрічей, за нагадуванням по електронній пошті та смс, генерування рахунків для сплати, онлайн оплата та інше.

Цей проект розвивається зараз і буде розвиватись далі.

СПИСОК ДЖЕРЕЛ

1. What is REST? [Електронний ресурс]. — Режим доступу: <http://www.restapitutorial.com/lessons/whatisrest.html>.
2. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF [Електронний ресурс]. — 2012. — Режим доступу: <https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>.
3. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. / К. Джонс — Перше, 2015. — 368 с
4. Anthony Gore — Full-Stack Vue.js 2 and Laravel 7 [Електронний ресурс]. — 2020. — Режим доступу: <https://bit.ly/2OEODzR>.
5. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2017. — 608 с.
6. MySQL. Довідник. MySQL AB. — М: «Вільямс», 2016 — 521 с.
Скотт Хокінс. Адміністрування веб-сервера Apache і керівництво по електронній комерції. / С. Хокінс. — М.: «Вільямс», 2001. — 336 с.
7. Л. Аткинсон, З. Сураскін. PHP7. Бібліотека професіоналу. / Л. Аткинсон, З. Сураскін — М. : «Вільямс», 2020 — 548 с.
8. Хокинс С. Администрирование Web-сервера Apache / С. Хокинс — М.: Вильямс, 2001. — 336 с.
9. Болье А. — Learning SQL [Електронний ресурс]. — 2005. — Режим доступу: <http://shop.oreilly.com/product/9780596007270.do>.