

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

СЕНИК ІВАН ОЛЕКСАНДРОВИЧ

Допускається до захисту:

завідувач кафедри
інформаційних технологій,
канд. техн. наук, доцент
_____ Т. В. Нескородева
« _____ » _____ 2021р.

**РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ДЛЯ
КАВ'ЯРНІ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Римар П. В., старший викладач
кафедри інформаційних технологій

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2021

АНОТАЦІЯ

Сеник І. О. Розробка автоматизованої системи управління для кав'ярні. Спеціальність 122 «Комп'ютерні науки», освітня програма «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2021.

Кваліфікаційна (бакалаврська) робота присвячена розробці автоматизованої системи управління для кав'ярні. За допомогою мови програмування Python та фреймворка Django розроблено веб – додаток, який використовується у кав'ярні. Процеси які були автоматизовані: облік проданих товарів, кількість відвідувачів для збору статистичної інформації, кількість товару з терміном придатності, формування замовлення товару.

Ключові слова: кав'ярня, Django, Python, веб-додаток.

Сторінок 44, рис 35, джерел 25.

ABSTRACT

SENYK Ivan. Development of an automated control system for a coffee shop. Specialty 122 «Computer Science», Educational program «Modern information technologies and programming», Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

Qualification (bachelor's) work dedicated to the development of an automated control system for a coffee shop. Using the Python programming language and the Django framework, a web application has been developed that is used in coffee shops. Processes that have been automated: accounting for goods sold, the number of visitors to collect statistical information, the number of expired goods, the formation of orders for goods.

Keywords: coffee shops, Django, Python , web-application.

Pages 44, Figure 35, sources 25.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ АВТОМАТИЗОВАНИХ СИСТЕМ ДАНОГО НАПРЯМУ	6
1.1 Постановка задачі	6
1.2 Огляд існуючих аналогів.....	6
Висновок до розділу 1	12
РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ	13
2.1 Kanban-дошка	13
2.2 Мова програмування PYTHON	15
2.3 Архітектурний шаблон MVT	17
2.4 Фреймворк Django	18
2.5 База даних PostgreSQL.....	21
2.6 Visual Studio Code	23
2.7 Pandas	25
2.8 Бібліотека NumPY	26
2.9 Фреймворк Bootstrap.....	26
2.10 Бібліотека D3.JS	28
2.11 Ajax	29
Висновок до розділу 2	30
РОЗДІЛ 3. РОЗРОБКА WEB – ДОДАТКУ ДЛЯ КАВ'ЯРНІ.....	31
3.1 Схема бази даних	31
3.2 Процес формування замовлення	32
Висновок до розділу 3	37
ВИСНОВОК.....	38
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	40

ВСТУП

Кав'ярні займають вагому частку в загальній сукупності закладів громадського харчування. На сьогоднішній день кав'ярні є одним із перспективних напрямів розвитку ресторанного бізнесу та мають вплив на економіку України. Визначено, що поширення кавової культури та освіченості споживачів у даній сфері робить кавовий бізнес перспективним напрямком діяльності для українських підприємців. Виявлено, що на сьогоднішній день поняття кав'ярні набуває нового значення – це заклад ресторанного бізнесу, де населенню пропонується широкий вибір кавових напоїв високої якості, різноманітних сортів кави як традиційних, так і альтернативних методів приготування та повна поінформованість бариста з приводу походження, рівня та свіжості обсмаження кавового зерна, а також тонкощів смаку кожного з варіантів. Обґрунтовано необхідність для закладів даного типу концентрувати увагу не тільки на поточних проблемах, а й розробляти довгострокову стратегію для подальшого розвитку. Кав'ярні сьогодні – це заклади ресторанного бізнесу, де кава являється особливим продуктом і готується на якісно новому рівні з індивідуальним підходом до кожного клієнта. Заклади, що позиціонують себе як кав'ярні – стають невід'ємною частиною індустрії та потребують подальших впроваджень з приводу утримання конкурентних позицій на довгострокову перспективу. [1]

Сьогодні в Україні спостерігається справжній кавовий бум. У різних містах відкривається все більше нових кавових закладів. З огляду на любов українців до хорошої та якісної кави в затишній атмосфері, в якій приємно відпочити після робочого дня – це хоч і невеликий, але досить прибутковий вид бізнесу.[2]

Автоматизація процесів управління в кав'ярні є актуальним завданням для адміністраторів і власників закладів сфери. Без системи управління та обліку неможливо уявити жоден сучасний заклад. Система грає важливу роль в оперативності та зручності управлінні кав'ярнею. Для точного обліку в кав'ярні

завжди потрібно знати, скільки товару є на складі, скільки напівфабрикатів або заготовок товару для випічки або реалізації вже є на складі та який їх термін придатності. Завдяки використанню автоматизованої системи завжди можна переглянути актуальні дані про продажі, обсяги та складський залишок.[3]

Мета бакалаврської роботи: вивчення методики проектування web-додатків на мові програмування Python з використанням фреймворка Django.

Задачами дослідження є:

1. огляд існуючих програм за даною тематикою, виявлення їх переваг та недоліків;
2. вивчення спеціалізованого середовища розробки (Visual Studio Code) для проектування web-додатку;
3. розробка автоматизованої системи управління для обліку роботи процесів у кав'ярні.

Об'єктом для дослідження є мова програмування «Python» та фреймворк «Django» а саме, наскільки зручно на їх основі проектувати web-додатки.

Предметом для дослідження є механізм розробки web-додатку для автоматизованої системи управління із застосуванням вищенаведених інструментів та технологій.

Під час виконання роботи належить вивчити фреймворк «Django» в контексті застосування з мовою програмування Python, створити базу даних для збереження необхідної інформації в середовищі Postgresql, а також навчитись застосовувати архітектурний шаблон «MVT» під час проектування web-додатку.

Структура роботи: вступ, три розділи, висновок, список літературних джерел кількістю 25 найменувань та 35 рисунків. Загальний об'єм роботи 44 сторінки.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ АВТОМАТИЗОВАНИХ СИСТЕМ ДАНОГО НАПРЯМУ

В даному розділі наведена постановка задачі, а також проведено огляд існуючих програм за даною тематикою.

1.1 Постановка задачі

Необхідно розробити web-додаток для автоматизації процесів кав'ярні. Логіка web-додатку побудована на основі використання Kanban-дошки. Будуть автоматизовані наступні процеси: облік проданих товарів, кількість відвідувачів для збору статистичної інформації, кількість товару з терміном придатності, формування замовлення товару.

1.2 Огляд існуючих аналогів

На сьогоднішній день на ринку є багато web-сервісів для автоматизації процесів роботи кав'ярень. Проте таких, які використовують Kanban-дошку в своїй основі – не має.

Розглянемо аналогічні автоматизовані системи.

Система «POSTER».[4] Вона представляє собою касу для кав'ярні. Користувачу не потрібно купувати моноблоки та інше спеціальне обладнання, щоб встановити програмне забезпечення для цього. Poster запускається на звичайних планшетах, які підходять за системними налаштуваннями.

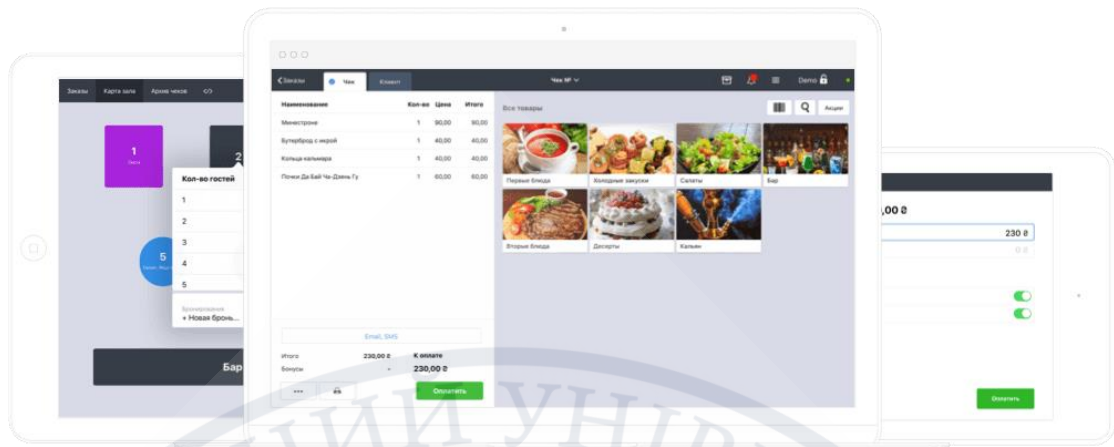


Рисунок 1.1 – Головний екран

Особливостями даного додатку є:

1. Зручна та зрозуміла навігація по меню.

Можна назначити окрему картинку чи фото для кожного напою та категорії.

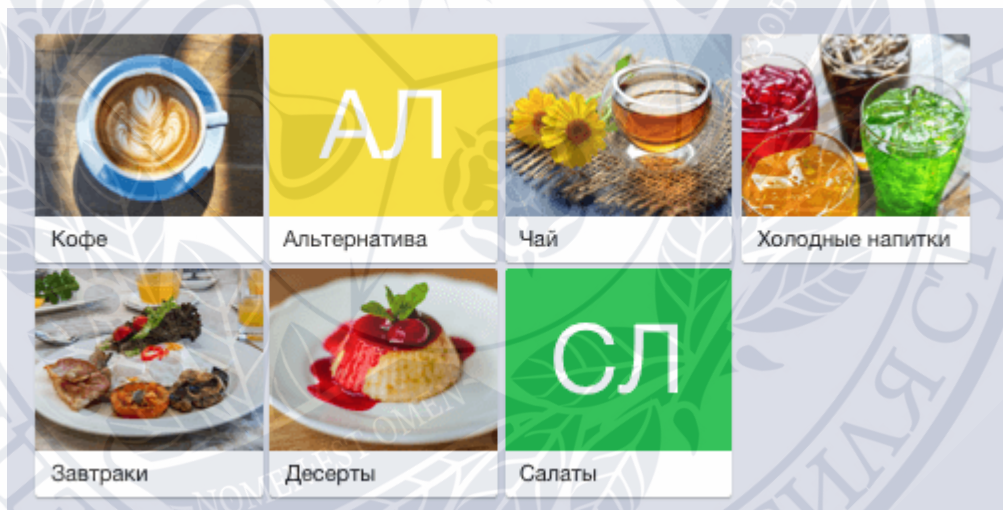


Рисунок 1.2 – Меню

2. Розрахунок біля столу або на касі.

Не важливо, як ви плануєте обслуговувати гостей: на касі чи виносити чек до столику. Якщо ви бажаєте, щоб бариста оформляв замовлення на столики, можна включити даний режим в терміналі додатку.

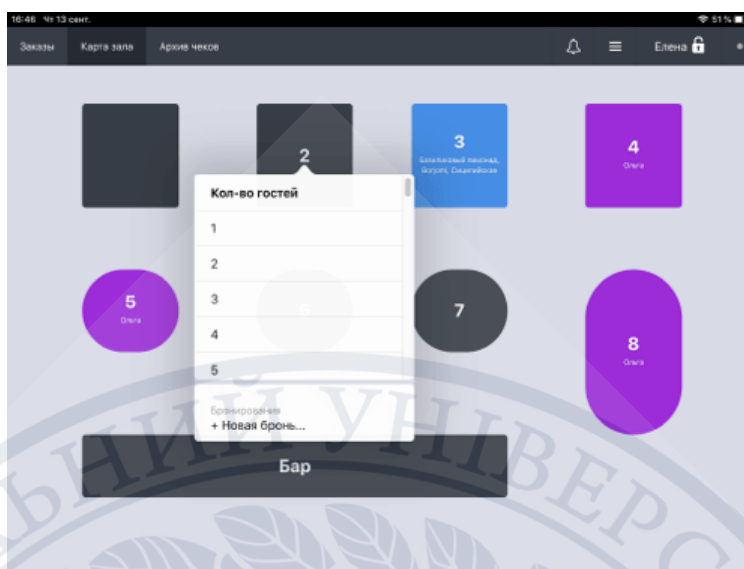


Рисунок 1.3 – Занятість столів

3. Модифікатори.

Використання модифікаторів до напоїв. Наприклад, якщо гість хоче лате не на звичайному молоці, а на безлактозній альтернативі або просить напій без кофеїну.

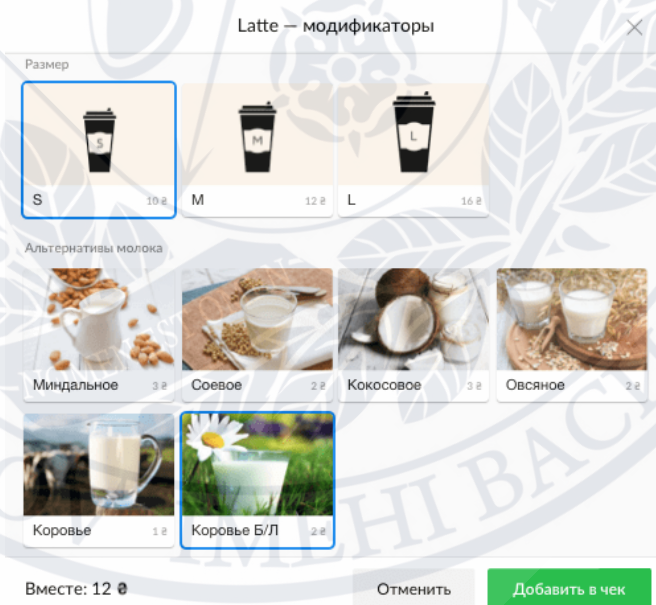


Рисунок 1.4 – Модифікатори

4. Вся інформація на вашому ноутбукові.

Адмін-панель знаходиться в браузері, де можна додати поставку, відкоригувати меню, перевірити залишки товару.



Рисунок 1.5 – Адмін панель

5. Зручно слідкувати за витратами.

Можна щоденно дивитись на звіти, щоб завжди бути в курсі актуальних продажів, витрат та прибутку. Можна бачити витрати на поставки продуктів, неправильні розміри порцій та інше.

Отчет по категориям Экспорт 8 августа – 8 сентября 8 авг – 8 сент

Все счета по месяцам

Категория	авг 2020	сент 2020
Актуализация	-511,52₴	-116,52₴
Поставки	-51 548,92₴	-23 323,69₴
Кассовые сменн	-52 317,00₴	-13 924,00₴
Всего расходов	-104 377,44₴	-37 364,21₴
Актуализация	+536,00₴	+13,00₴
Кассовые сменн	+228 747,00₴	+52 426,00₴
Всего доходов	+229 283,00₴	+52 439,00₴
Финансовый результат	+124 905,56₴	+15 074,79₴

Рисунок 1.6 – Звіт

6. Інвентаризація.

В Poster можна провести як повну так і часткову інвентаризацію складу. Якщо потрібно перевірити залишки деякого продукту, просто можна перейти на вкладку перевірки запасів в адмін-панелі.

Инвентаризации 77

Быстрый поиск Склад + Фильтр

Экспорт Печать За все время

Склад	Дата начала	Дата конца	Сумма	Статус	
Склад Кухня	18 мая 2017, 09:10		98,00 ₴	Текущая	Открыть ***
Склад Бар	20 апреля 2017, 10:28		751,00 ₴	Текущая	Открыть ***
Склад Кухня	12 августа 2015, 19:07	18 мая 2017, 09:10	483,98 ₴	Проведенная	Открыть
Склад Бар	4 августа 2016, 00:20	20 апреля 2017, 10:28	-1,42 ₴	Проведенная	Открыть
Склад Бар	9 августа 2013, 16:30	4 августа 2016, 00:20	-70,00 ₴	Проведенная	Открыть
Склад Кухня	5 февраля 2015, 13:10	12 августа 2015, 19:07	1 043,84 ₴	Проведенная	Открыть
Склад Кухня	9 августа 2013, 16:30	5 февраля 2015, 13:10	-212,36 ₴	Проведенная	Открыть
Склад Кухня	18 мая 2017, 09:10		98,00 ₴	Текущая	Открыть ***
Склад Бар	20 апреля 2017, 10:28		751,00 ₴	Текущая	Открыть ***
Склад Кухня	12 августа 2015, 19:07	18 мая 2017, 09:10	483,98 ₴	Проведенная	Открыть
Склад Бар	4 августа 2016, 00:20	20 апреля 2017, 10:28	-1,42 ₴	Проведенная	Открыть
Склад Бар	9 августа 2013, 16:30	4 августа 2016, 00:20	-70,00 ₴	Проведенная	Открыть

Рисунок 1.7 – Інвентаризація

Наступний аналог – система «SmartTouch POS».[5]

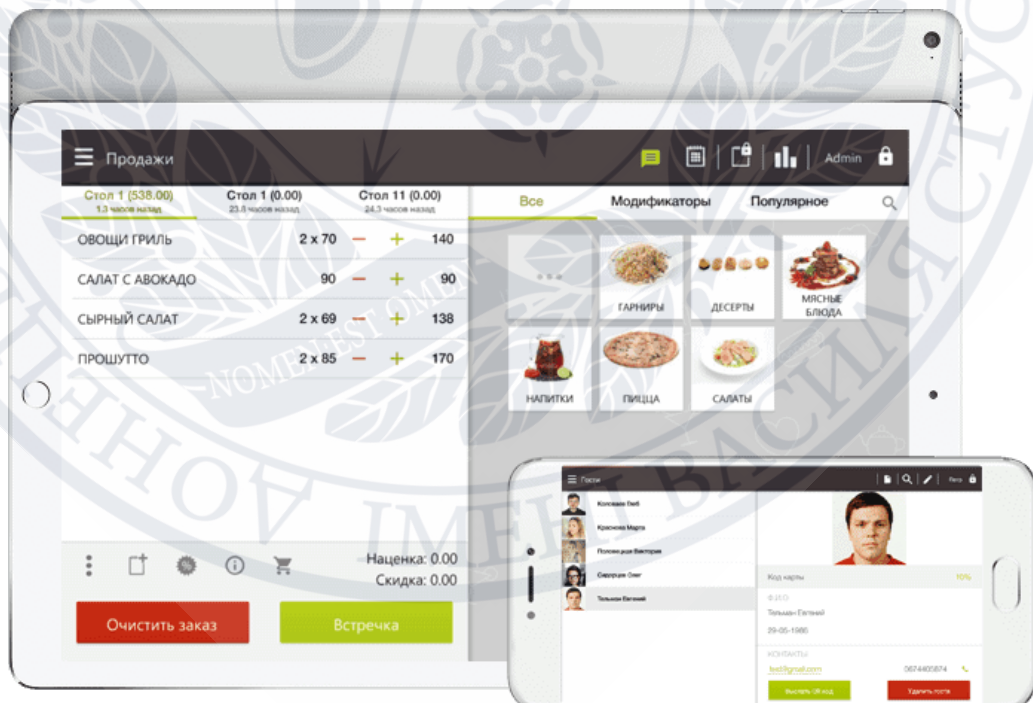


Рисунок 1.8 – Головний екран «SmartTouch POS».

Особливостями даного сервісу є:

1. Продаж на залі, доставці, виїзній торгівлі.
2. Обмін даними з кухнею, складом.
3. Всі види оплати, інтеграція з касовою технікою.
4. Автоматичні звіти керуючому.



Рисунок 1.9 – Звітність в «SmartTouch POS»

«SmartTouch Boss» – контроль продажів з телефону. Додаток, який відправляє звіт по доходам, відвідуваність, середній чек, відкриті та закриті замовлення, продаж по товарам та видам оплати.



Рисунок 1.10 – Контроль з продажів

«SmartTouch eMenu» для гостей кав'ярні або ресторану Додаток, в якому є фото і опис продукції, а також актуальні ціни.

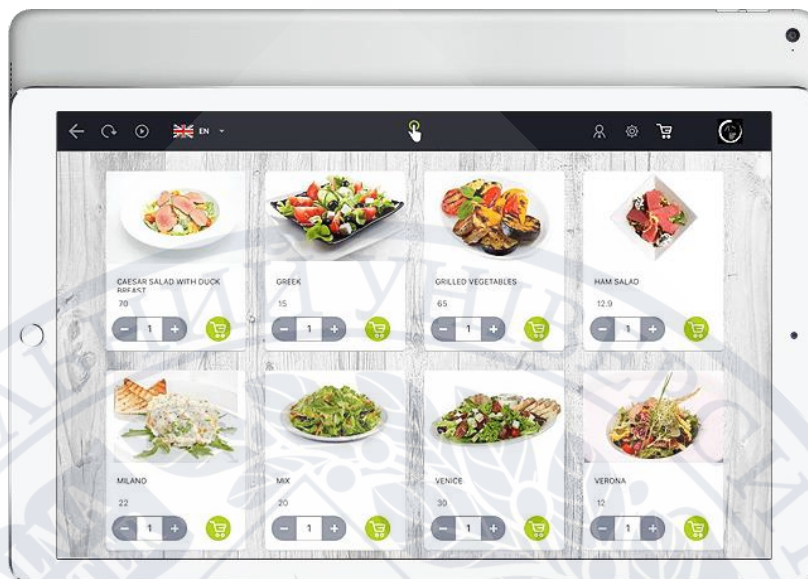


Рисунок 1.11 – Меню для гостей

Висновок до розділу 1

В цьому розділі було описано постановку задачі бакалаврської роботи, а також проведений огляд існуючих додатків та сервісів за даною тематикою, також були окреслені їх особливості. Наступний розділі буде присвячений вибору інструментів для розробки.

РОЗДІЛ 2

ВИКОРИСТАНІ ТЕХНОЛОГІЇ

Даний розділ присвячений опису використаного стеку технологій, які використовувалися під час проектування web-додатку.

2.1 Kanban-дошка

Дошка Kanban – це інструмент управління Agile-проектами, який допомагає наочно уявити завдання, обмежити обсяг незавершеної роботи і домогтися максимальної ефективності.[8] Вона може допомогти командам Agile і DevOps впорядкувати повсякденну роботу. За допомогою карток і стовпців на дошці Kanban команди з технічних питань і сервісні команди можуть зрозуміти, який обсяг роботи слід взяти на себе і виконати цей обсяг, дотримуючись принципів безперервного вдосконалення.[7]

Як влаштований Kanban в проектах

Кожен проєкт містить свій план процесу робіт. Насамперед аналізується план і поділяється дошка на стовпці, які відображають етапи.[6]

Kanban-картками називаються завдання, які перетікають в інші стовпці в залежності від їх стану та рухаються по потоку . За допомогою Kanban-дошки легко вести кілька проєктів одночасно, використовуючи картки різних кольорів.

Складові дошки Kanban

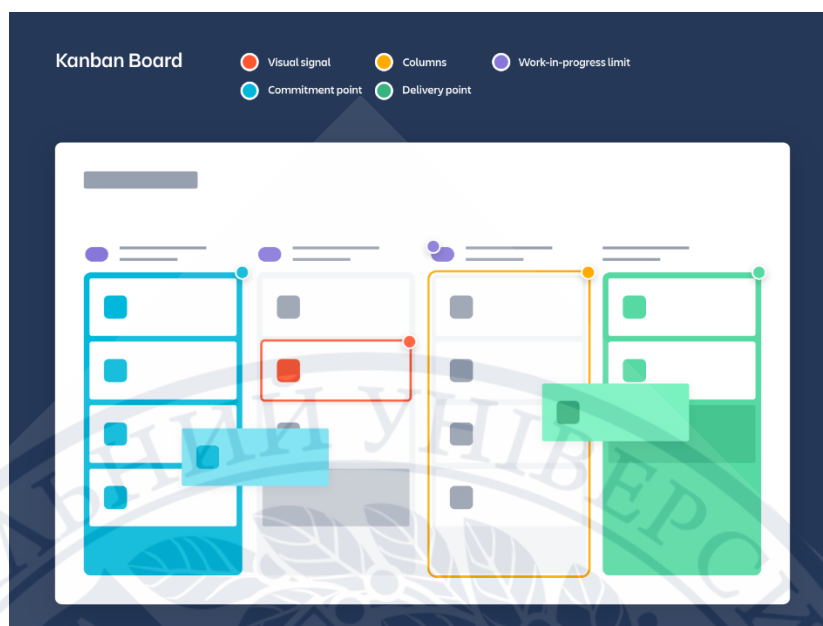


Рисунок 2.1 – Kanban-дошка

1. Видимі сигнали. Першими на дошці Kanban кидаються в очі картки. Kanban-команди виносять записи про всі проекти і робочі завдання на картки; одна картка, як правило, відповідає одному проекту або робочій задачі. Для Agile-команд кожна картка позначає одну призначену для користувача історію. Побачивши ці сигнали на дошці, учасники команди і зацікавлені сторони зможуть легко зрозуміти, над чим працює команда.
2. Стовпці. Ще однією відмінною рисою дошки Kanban є стовпці. Вони символізують конкретні дії, які в сукупності складають «робочий процес». Картки переміщуються по робочому процесу до стадії завершення. Робочі процеси можуть бути простими і складатися лише зі стовпців «Має бути зроблено», «В процесі» і «Завершено», а можуть бути набагато складнішими.
3. Обмеження незавершеної роботи (WIP). Обмеження WIP – це максимальна кількість карток, що може перебувати в одному стовпці одночасно. Якщо для стовпця вибрано обмеження WIP, що дорівнює 3, то в ньому не може бути більше трьох карток. Коли кількість карток в стовпці досягає максимуму, команда повинна зосередити зусилля на цих картках і передати їх далі, щоб на цю стадію робочого процесу могли зробити нові

картки. Обмеження WIP потрібні, щоб виявляти проблемні місця в робочому процесі і домагатися максимальної швидкості роботи. Обмеження WIP допомагають на ранніх етапах зрозуміти, чи не взяла команда на себе занадто багато завдань.

4. Точка прийняття зобов'язань. На дошці у Kanban-команд часто присутній беклог. Клієнти і учасники команди вносять в нього ідеї щодо проектів, до яких команда може звернутися, коли буде готова. У точці прийняття зобов'язань команда вибирає ту чи іншу ідею, після чого починається робота над проектом.
5. Точка поставки продукту. Точка поставки продукту знаменує завершення робочого процесу команди Kanban. Багато команд беруть за точку поставки продукту момент, коли продукт або сервіс передаються в розпорядження клієнта. Мета команди – якнайшвидше перенести картки з точки прийняття зобов'язань в точку поставки продукту. Час, за яке картка проходить з однієї точки в іншу, називається часом виконання. Kanban-команди постійно вдосконалюються, прагнучи звести час виконання до мінімуму.[7]

2.2 Мова програмування PYTHON

Python – це проста у вивченні мова програмування. Вона дає дієві структури даних, а також простий, але ефективний підхід до об'єктно-орієнтованого програмування. Його зручний для читання синтаксис і динамічна типізація разом з тим, що він інтерпретується, роблять його довершеною мовою програмування для написання сценаріїв та швидко - ефективною розробки додатків в різних областях і на більшості платформ. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять його принадливим для швидкої розробки програм, а також як засіб з'єднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду.

Переваги PYTHON:

- Чистий синтаксис (для виділення блоків слід використовувати відступи).
- Стандартний дистрибутив має багато модулів (включаючи модуль для розробки графічного інтерфейсу).
- Можливо використовувати Python, в діалоговому режимі (дуже корисне для експериментування та розв'язування легких задач).
- Стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки яке зветься IDLE та написано мовою Python.
- Зручний для розв'язку математичних проблем (має засоби роботи з комплексними числами, оперує з цілими числами будь – якої величини, у діалоговому режимі може використовуватись як калькулятор).

Особливості PYTHON:

1. Вільний та відкритий

Python – це приклад вільного та відкритого програмного забезпечення – FLOOS (Free/Libre and Open Source Software). Простіше кажучи, користувач має можливість вільно поширювати копії даного програмного забезпечення, вносити зміни, читати його вихідні тексти, а також застосовувати його частини в своїх програмах.

2. Мова високого рівня

При написанні програми на Python розробнику ніколи не доведеться відволікатися на такі низькорівневі деталі, як управління пам'яттю, яку використовує програма.

3. Портативний

Всі програми, яуї написані з використанням Python, зможуть запускатися на будь-який з платформ без будь-яких змін, якщо тільки розробник уникав використання системно-залежних функцій.

4. Інтерпретований

Python не потребує компіляції в бінарний код. Програма просто виконується з початкового тексту. Python, сам перетворює вихідний текст в деяку проміжну форму, так звану байткод, а потім переводить його на машинну мову і запускає.

5. Об'єктно – орієнтований

В об'єктно-орієнтованих мовах програмування програми будуються на основі об'єктів, які об'єднують в собі дані і функціонал.

6. Великі бібліотеки

Стандартна бібліотека Python просто величезна. Вона може допомогти у вирішенні найбільш різноманітних завдань, пов'язаних з використанням регулярних виразів, генеруванням документації, перевіркою блоків коду, розпаралелюванням процесів, базами даних, веб-браузерами, криптографією, GUI (графічний інтерфейс користувача) і іншими системними речами.[9]



Рис 2.2 – Python

2.3 Архітектурний шаблон MVT

MVT – Model-View-Template або модель-уявлення-шаблон. Складові MVT можна використовувати незалежно один від одного.[10]

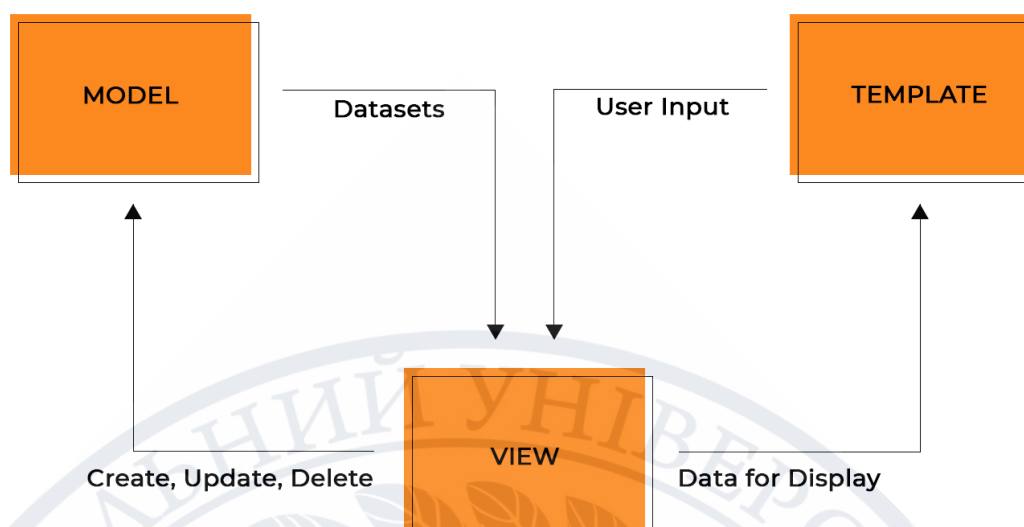


Рисунок 2.3 – Модель MVT

View – вирішує три завдання: приймає HTTP-запити, реалізує бізнес-логіку, певними методами і властивостями, відсилає HTTP-відповідь у відповідь на запити.

Models – відповідає за бізнес-логіку, методи, властивості та інші елементи, які пов'язані з процедурами над даними. За допомогою моделі розробники можуть: створювати, читати, оновлювати та видаляти об'єкти в базі даних.

Templates[10] – це текстовий файл, який визначає структуру або розмітку сторінки (наприклад HTML-сторінки) з полями для підстановки, які використовуються для виведення актуального вмісту. View може динамічно створювати HTML-сторінки, використовуючи HTML-шаблони і заповнюючи їх даними з моделі (model). Шаблон може бути використаний для визначення структури файлів будь-яких типів, не обов'язково HTML.[10]

2.4 Фреймворк Django

Django – безкоштовний і вільний фреймворк для веб-додатків, написаний на Python. Фреймворк – це набір компонентів, які допомагають розробляти веб-сайт швидко і просто.

Кожен раз при розробці веб-сайтів потрібні схожі компоненти: спосіб аутентифікувати користувачів (вхід, вихід, реєстрація), панель управління

сайтом, форми, інструменти для завантаження файлів. Фреймворк пропонує готові шаблони для використання.[11] Django бере на себе більшу частину веб-розробки. Він є безкоштовним фреймворком із відкритим вихідним кодом, має активну спільноту, відмінну документацію та безліч варіантів як безкоштовної, так і платної підтримки.[12] В Django реалізований потужний движок шаблонів і власна мова розмітки. Шаблони представляють собою файли з HTML-кодом, завдяки яким візуалізуються дані. Файл може бути статичним або динамічним. Шаблони не містять бізнес-логіки, тому вони тільки відображають дані.[13]

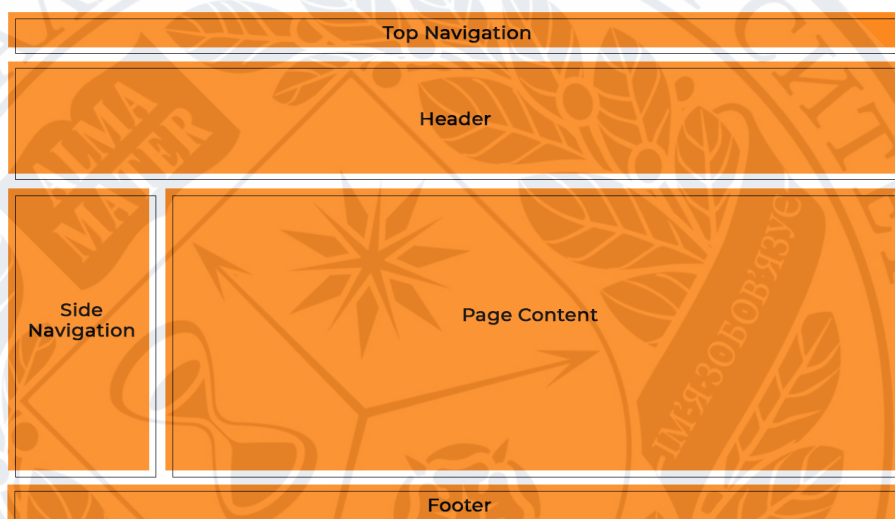


Рисунок 2.4 – Шаблон сторінки

Фреймворк Django справляється великою кількістю завдань і підвищеними навантаженнями. Його використовують для створення:

1. CRM-систем.
2. CMS.
3. Комунікаційних платформ.
4. Сервісів бронювання номерів.
5. Платформ управління документообігом.

Створення алгоритмічний генераторів, систем верифікацій, платформ для електронних розсилок, систем фільтрації з складними параметрами та динамічними правилами, а також створення машинного навчання – усе це можна реалізувати за допомогою фреймворка Django.[13] Також даний фреймворк

зручний при розробці різнотипних веб – сайтів, включаючи системи управління контентом, новинний сайт, wiki до соціальних мереж. Він може доставляти контент у форматах HTML, JSON, XML, RSS канали та інші, а також працює майже у всіх клієнтських середовищах. [12]

Уникнення багатьох поширених помилок безпеки, надання безпечного керування обліковими записами і паролями користувачів, уникання загальних помилок таких як: розміщення про сеанс в файли cookie, які містять тільки ключ а фактичні дані зберігаються в базі даних або зберігання паролів замість хеш – пароля Django допомагає розробникам уникнути цього.[12] Адмін – панель Django генерується автоматично при створенні програми, непотрібно кожного разу створювати її вручну.[13]

Стандартну консоль управління Django можна вдосконалити з допомогою сторонніх додатків і адаптувати під потреби свого проекту. Крім цього, є можливість налаштування інтерфейсу стандартної адмін – панелі. За допомогою плагінів функціональність Django розширюється. Плагіни - це програмні модулі, завдяки яким, потрібна інформація на сайт додається дуже швидко. Плагіни при необхідності можна відключати або замінювати, щоб прилаштувати їх до поточних потреб проекту. Також в фреймворку реалізовано об'єктно-реляційне відображення (ORM), завдяки якому забезпечується взаємодія додатка з базами даних (БД). ORM автоматично передає дані з БД, наприклад, PostgreSQL або MySQL, в об'єкти, які використовуються в коді програми.

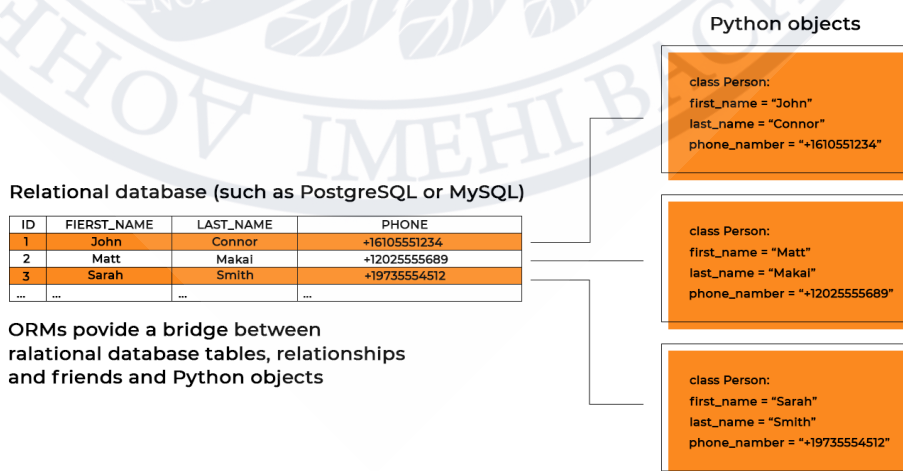


Рисунок 2.5 – Схема роботи ORM

ORM прискорює розробку прототипів і готових веб-додатків на Django. Девелоперу навіть не потрібно знати мову, яка використовується для взаємодії з базами даних. Завдяки ORM можливо швидко перемикається між базами даних з мінімальними змінами коду. Наприклад, ви можете використовувати SQLite на локальному сервері, а потім переключитися на MySQL на production-сервері. Однак для мінімізації помилок краще використовувати одну базу даних під час розробки.[13] Django написаний на Python, тому він являється кросплатформеним фреймворком. Відповідно користувач може запускати додатки на багатьох версіях Linux, Windows, Mac OS X, Django добре підтримується багатьма веб-хостингами, які часто надають певну інфраструктуру і документацію для розміщення сайтів Django.[12]

2.5 База даних PostgreSQL

PostgreSQL – це СКБД з відкритим вихідним кодом, розроблена всесвітньої командою добровольців. PostgreSQL не контролюється будь-якою корпорацією або іншим приватним особою, і вихідний код доступний безкоштовно.[14]



Рисунок 2.6 – PostgreSQL

Базова характеристика об'єктно-реляційної бази даних – це підтримка об'єктів і їх поведінки. Вона включає в себе такі типи даних: функції, операції,

домени та індекси. PostgreSQL являється неймовірно гнучким і надійним. Серед іншого, він вміє створювати, зберігати та видавати складні структури даних. [15]

Переваги PostgreSQL:

- Підтримка БД необмеженого розміру.
- Потужні і надійні механізми транзакцій та реплікацій.
- Розширювана система вбудованих мов програмування і підтримка завантаження C-сумісних модулів.
- Наслідувальність.
- Легка розширюваність.

Поточні обмеження PostgreSQL:

- Обмеження на максимальний розмір бази даних відсутні.
- Обмеження на кількість записів в таблиці відсутні.
- Обмеження на кількість індексів в таблиці відсутні.
- Допустимий розмір таблиці – 32 Тбайт.
- Допустимий розмір запису – 1,6 Тбайт.
- Допустимий розмір поля – 1 Гбайт.
- Допустимих полів в записі 250-1600.

Особливості PostgreSQL:

Функції в PostgreSQL є блоками коду, які виконуються сервером. Для написання переходів та циклів необхідно застосовувати спеціальні розширення. Власне SQL і вимагає використання деяких мовних розширень. Для опису функцій можна скористатися довільною мовою програмування. Функції виконуються як для користувача зі звичайними правами, так і з правами адміністратора. Також функції порівнюють зі збереженими процедурами, але це 2 різних поняття за змістом.

PostgreSQL дозволяє використання тригерів, які ініціюються DML - операціями. Виконання множинних тригерів відбувається за алфавітом.

PostgreSQL містить в собі набір правил для розробки не тільки DML – операцій, а також операцій вибірки. Ці правила виконуються під час розбору

запиту. За їх допомогою можна змінити поведінку системи під час виконання SQL – операцій.

PostgreSQL містить індекси: хеш, R-дерево, B-дерево, GIN, GiST. За потреби розробник може створити індекси нових типів.

Використовуючи механізм MVCC можна одночасно створювати декілька версій БД різними користувачами. Використовуючи цей підхід не має необхідності в операції блокування читання та дотримуються вимоги ACID.

Для власних потреб розробник може розширити функції PostgreSQL, а саме: додавати нові домени та типи даних , агрегатні функції, індекси , перетворення типів, а також перевизначати існуючі оператори.

Таблиці мають властивість наслідування. Вони можуть наслідуватись від інших таблиць (батьківських). Це дозволяє даним брати участь в запитах до батьківських таблиць.

Використання в веб-проектах

Під час розробки простих веб – додатків також може використовуватись PostgreSQL, хоча, наприклад, MySQL / MariaDB використовується частіше ніж інші СУБД. Але для розробки складних веб – додатків PostgreSQL випереджає їх по частоті використання. [16]

2.6 Visual Studio Code



Рисунок 2.7 – Visual Studio Code

Visual Studio Code – безкоштовне IDE для розробки програмного забезпечення.

Особливістю даної IDE є те, що за допомогою опції «Пошук», можна знаходити необхідні параметри в коді. А також можна користуватись функцією «Заміна».

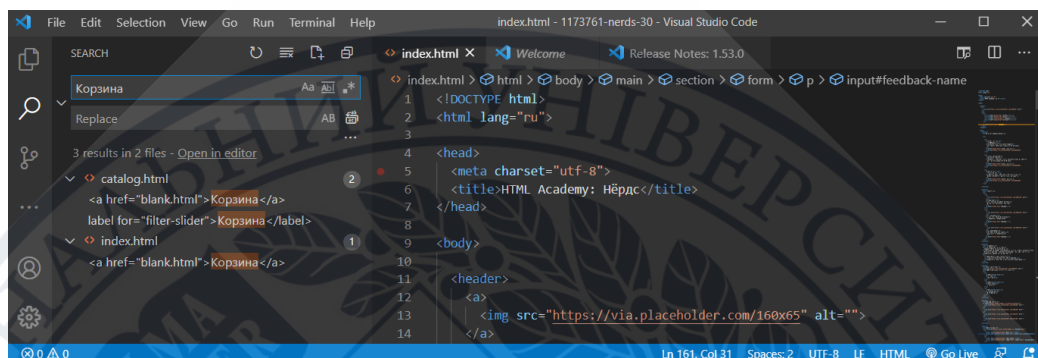


Рисунок 2.8 – Пошук в Visual Studio Code

GIT

VS Code з коробки підтримує роботу з системою контролю версій Git. Можна виконати базові операції і подивитися, як змінилися файли з моменту останнього коміта.

Отладчик

VS Code відрізняє від конкурентів наявність вбудованого засобу для відладки коду. Для перевірки коду на помилки, можна встановити точку зупинки і дивитися за тим, що покаже програма. Також редактор міститься в собі вбудовану консоль, на неї може виводитися результат роботи, або повідомлення з помилками. Отладчик можна налаштувати під різні мови і різні завдання.

Розширення

Останній пункт активного меню дозволяє встановлювати розширення. В VS Code є розширення для більшості мов програмування.[17]

2.7 Pandas

Pandas – це високорівнева Python бібліотека для аналізу даних. В екосистемі Python, pandas є найбільш просунутою бібліотекою для обробки і аналізу даних.



Рисунок 2.9 – Pandas

Використовуючи Pandas, розробник може виконати п'ять типових кроків з обробки та аналізу даних, незалежно від походження даних – завантажити, підготувати, маніпулювати, моделювати та аналізувати.

Python з Pandas використовується в широкому спектрі областей, включаючи академічні та комерційні області, включаючи фінанси, економіку, статистику, аналітику і т.д.

Особливості Pandas:

- Швидкий і ефективний об'єкт DataFrame з індивідуальною індексацією за замовчуванням.
- Інструменти для завантаження даних в об'єкти даних в пам'яті з різних форматів файлів.
- Вирівнювання даних та інтегрована обробка відсутніх даних.
- Зміна форми і поворот наборів дат.
- Мітка нарізки, індексація і підмножина великих наборів даних.
- Стовпці зі структури даних можуть бути видалені або вставлені.

- Групування за даними для агрегації і перетворень.
- Висока продуктивність злиття і об'єднання даних.
- Функціональність часових рядів.[18]

2.8 Бібліотека NumPY

Одна з бібліотек ,яка міститься в складі мови програмування Python – це бібліотека NumPy. Вона використовується для роботи з масивами та матрицями великого розміру. Також вона додає високо рівневі математичні функції для роботи з цими масивами та матрицями[19]



Рисунок 2.10 – NumPY

За допомогою NumPy можна маніпулювати великими масивами і матрицями. Математичні алгоритми, які реалізовані на Python, працюють набагато повільніше ніж на C, Java, або Фортран. [20] Алгоритм, який складається з операцій для роботи з масивами та матрицями і реалізований з використанням NumPy працює так само швидко як аналогічний код наприклад у MATLAB.[21]

2.9 Фреймворк Bootstrap

Bootstrap – це відкритий і безкоштовний HTML, CSS і JS фреймворк, який використовується веб-розробниками для швидкої верстки адаптивного дизайну сайтів та веб-додатків.



Рисунок 2.11 – Bootstrap

Основна область його застосування – це фронтенд розробка веб – додатків та сайтів. Серед аналогічних систем фреймворк Bootstrap є найпопулярнішим. Для роботи цього фреймворка необхідно підключити до веб – сторінки набір CSS і JavaScript файлів. Після цього розробнику стануть доступні інструменти даного фреймворка: колоночна система , класи і компоненти.

Bootstrap складається з:

- інструментів для створення макета;
- класів для стилізації базового контенту: тексту, зображень, коду, таблиць;
- готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, модальних вікон, випадаючих списків, акордеонів, слайдерів, спливаючих підказок і ін;
- утилітних класів для вирішення традиційних завдань, які найбільш часто виникають перед веб-розробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, відступів.

Переваги фреймворка:

- можуть створювати якісну адаптивну верстку веб – розробники з мінімальним досвідом ;
- кросбраузерність і кросплатформеність;
- наявна кількість готових компонентів;

- можна змінити кількість колонок, кольору, радіус заокруглень, відступи між колонками;
- низький поріг входження;
- наявність величезної кількості спільнот і навчальних матеріалів.[22]

2.10 Бібліотека D3.JS

D3.js – це бібліотека JavaScript, що використовується для створення інтерактивних візуалізацій в браузері. Бібліотека D3.js дозволяє нам маніпулювати елементами веб-сторінки в контексті набору даних. Ці елементи можуть бути елементами HTML, SVG або Canvas і можуть вводитися, віддаватися або редагуватися відповідно до вмісту набору даних. Це бібліотека для маніпулювання об'єктами DOM. D3.js може бути цінним помічником у дослідженні даних, він дає контроль над поданням даних і дозволяє додати інтерактивність.



Рисунок 2.12 – D3.JS

Особливості:

- Надзвичайно гнучкий.
- Простий та швидкий у використанні.
- Підтримує великі набори даних.
- Декларативне програмування.

- Повторне використання коду.
- Має широкий спектр функцій генерації кривої.
- Пов'язує дані з елементом або групою елементів на html-сторінці.

D3.js є проектом з відкритим вихідним кодом і працює без будь-якого плагіна. Це вимагає дуже менше коду і дає такі переваги:

- Відмінна візуалізація даних.
- Модульність. Можна завантажити невеликий шматочок D3.js, який потрібен для використання. Немає необхідності завантажувати всю бібліотеку кожен раз.
- Легко побудувати компоненти діаграм.
- DOM маніпулювання.[23]

2.11 Ajax

AJAX – це спеціальна технологія, за допомогою якої запит до сервера відбувається без перезавантаження сторінки. Це дозволяє зменшити час відгуку і веб – додаток стає більш схожим на десктоп – додаток. Хоча в назві присутня буква X (від слова XML) його використання не є обов'язковим . Під даною технологією розуміють організоване за допомогою JavaScript будь – яке спілкування з сервером без перезавантаження сторінки.



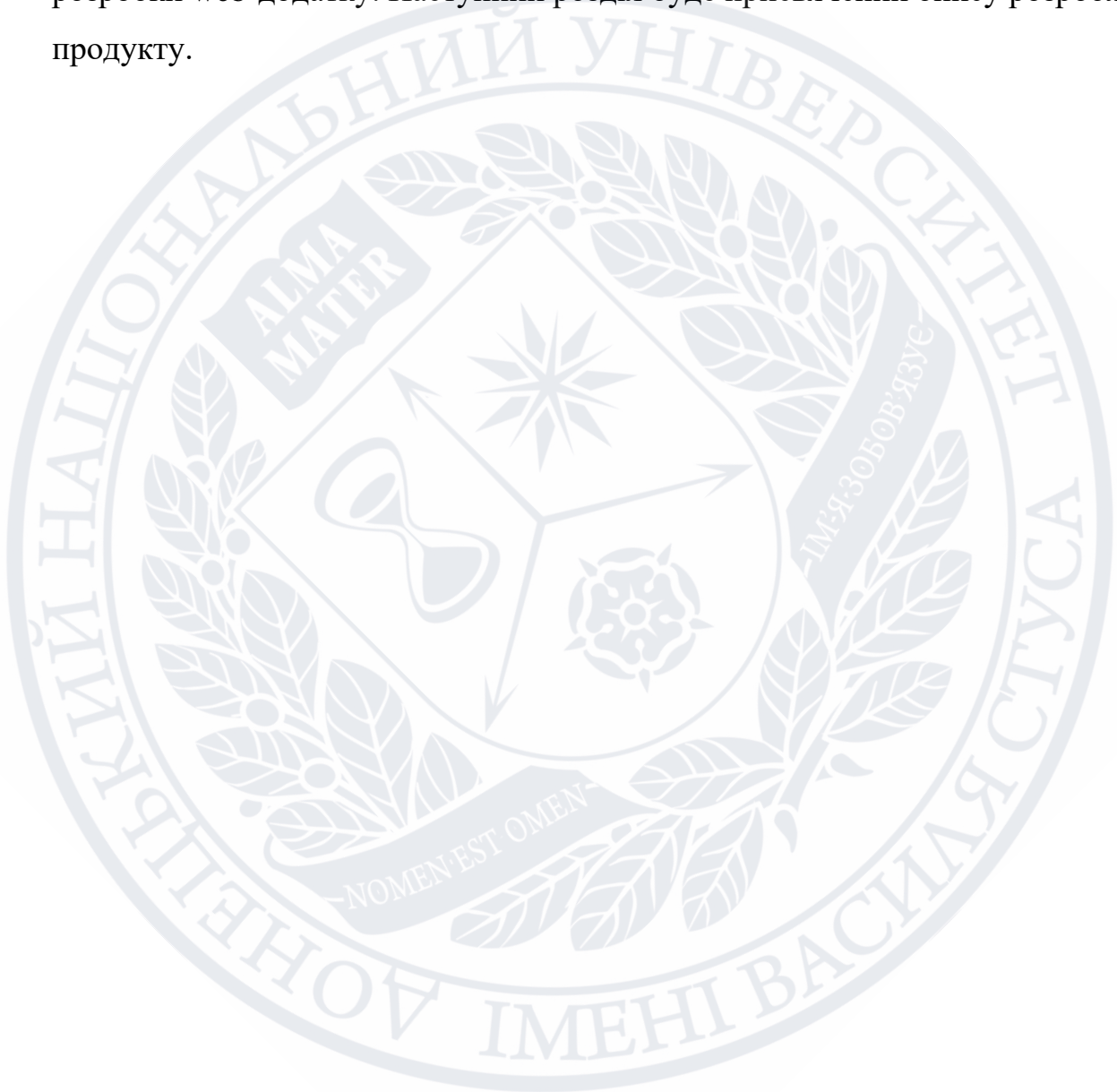
Рисунок 2.13 – AJAX

Найчастіше AJAX використовують для форм і кнопок, які виконують такі дії: підписатися, додати в кошик, оплатити і т.д. Це дозволяє створити зручний веб

– інтерфейс для взаємодії з користувачем та зручне використання веб – додатку. [25].

Висновок до розділу 2

В даному розділі був описаний стек технологій, які використовували для розробки web-додатку. Наступний розділ буде присвячений опису розробленого продукту.



РОЗДІЛ 3

РОЗРОБКА WEB – ДОДАТКУ ДЛЯ КАВ'ЯРНІ

Даний розділ присвячений опису розроблюваного веб-додатку для автоматизації процесів у кав'ярні.

3.1 Схема бази даних

На наступному рисунку наведена схема розробленої бази даних.

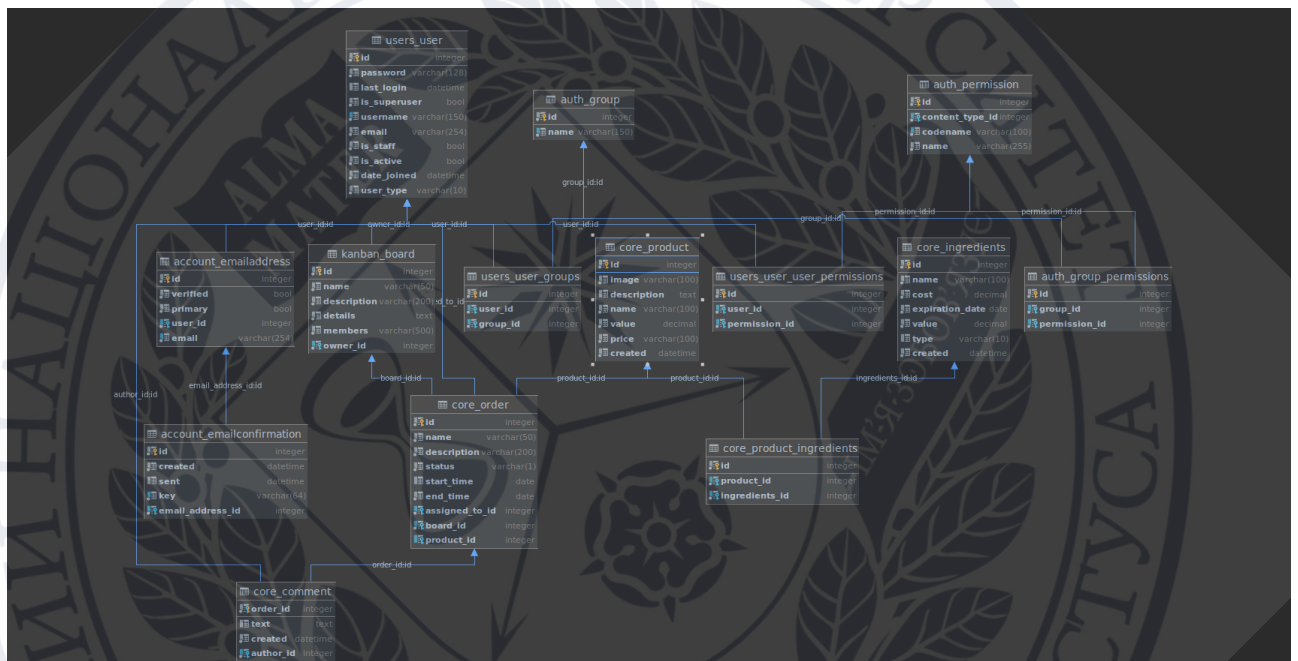


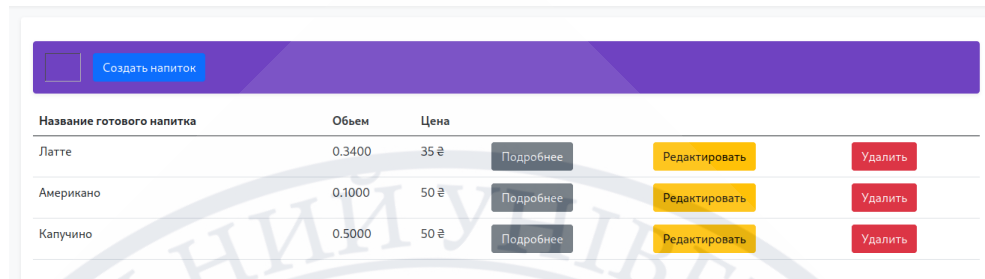
Рисунок 3.1 – Схема бази даних

На рис. 3.1 показана схема бази даних веб-додатку для кав'ярні, на якій показана взаємодія елементів бази даних між собою. База даних містить наступну інформацію:

- Про замовлення, які були зафіксовані за допомогою Kanban-дошки.
- Замовлення, які знаходяться в стадії виконання, а також ті, які вже виконано.
- Контактну інформацію про баристу.
- Інформація про поставлений товар: дата та час поставки, тип товару, одиниці виміру, кількість, вартість.

- Про інгредієнти для приготування напою.

3.2 Процес формування замовлення



Название готового напитка	Объем	Цена			
Латте	0.3400	35 ₴	Подробнее	Редактировать	Удалить
Американо	0.1000	50 ₴	Подробнее	Редактировать	Удалить
Капучино	0.5000	50 ₴	Подробнее	Редактировать	Удалить

Рисунок 3.2 – Перелік доступних напоїв

На рис. 3.2 наведено екран з переліком доступних напоїв. На цьому екрані можна побачити: назву напою, його об'єм та ціну. Можна додавати новий напій, редагувати складники або видаляти його.



Изображение товара*

Ингредиенты

- Кофейные зерна
- Молоко
- Сливки

Описание товара

Наименование товара*

Общий объем*

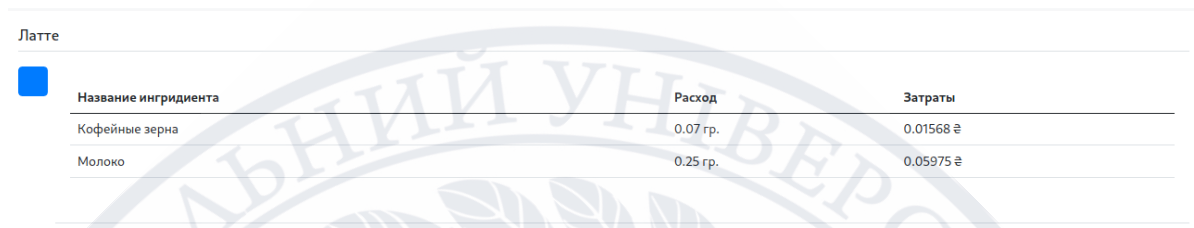
0

Цена*

Сохранить

Рисунок 3.3 – Додавання/ редагування напою

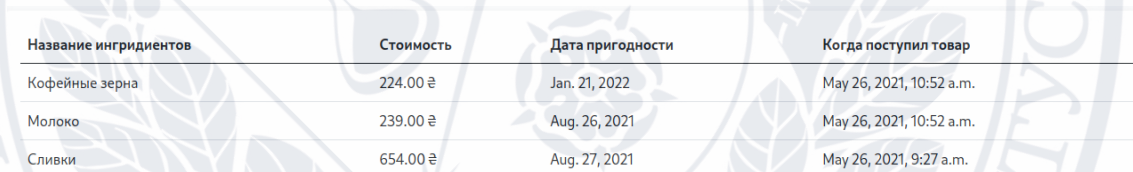
На рис. 3.3 показано екран додавання нового напою або редагування існуючого. При формуванні нового напою обираються його складники, назва, об'єм та ціна. При редагуванні можна змінити інгредієнти, або додати нові, також можна змінити об'єм та ціну.



Латте		
Название ингредиента	Расход	Затраты
Кофейные зерна	0.07 гр.	0.01568 €
Молоко	0.25 гр.	0.05975 €

Рисунок 3.4 – Складові напою

На рис. 3.4 вказані складові напою. Для прикладу обрано «Лате», в якому можна бачити назву інгредієнту, який використовується, які витрати продукції яка на складі береться, а також затрати в грошовому еквіваленті.



Название ингредиентов	Стоимость	Дата пригодности	Когда поступил товар
Кофейные зерна	224.00 €	Jan. 21, 2022	May 26, 2021, 10:52 a.m.
Молоко	239.00 €	Aug. 26, 2021	May 26, 2021, 10:52 a.m.
Сливки	654.00 €	Aug. 27, 2021	May 26, 2021, 9:27 a.m.

Рисунок 3.5 – Наявність товару на складі

На рис. 3.5 зображено, який товар є на складі, його ціна, дата та час, коли товар поступив на склад, а також термін його придатності.

Продукт	Цена	Время заказа	Время выполнения	
Латте	35 ₴	May 26, 2021	May 26, 2021	Подробнее
Американо	50 ₴	May 26, 2021	May 26, 2021	Подробнее
Латте	35 ₴	May 26, 2021	May 20, 2021	Подробнее
Капучино	50 ₴	May 26, 2021	May 20, 2021	Подробнее

Рисунок 3.6 – Виконані замовлення

На рис. 3.6 зображені замовлення, які були виконанні на Kanban – дошці. Вказано який саме напій було приготовано, його ціна, час замовлення, а також час виконання.

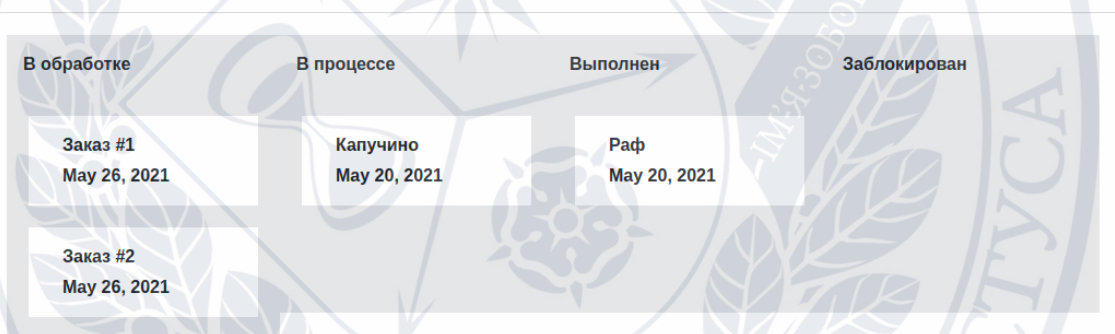


Рисунок 3.7 – Kanban – дошка

На рис. 3.7 зображена Kanban – дошка. На ній зображено 4 етапи виконання замовлення. «В обработке» – на цьому етапі показана інформація, яке замовлення було отримане на дошку і коли. «В процессе» – на другому етапі вказано, яке замовлення почало виконуватись. «Выполнен» – вказано, яке замовлення було виконано. «Заблокирован» – цей пункт призначено для того, якщо оплата була виконана картою і оплата не пройшла, тоді замовлення переходить в пункт «Заблокирован» до того моменту, поки оплата не пройде.

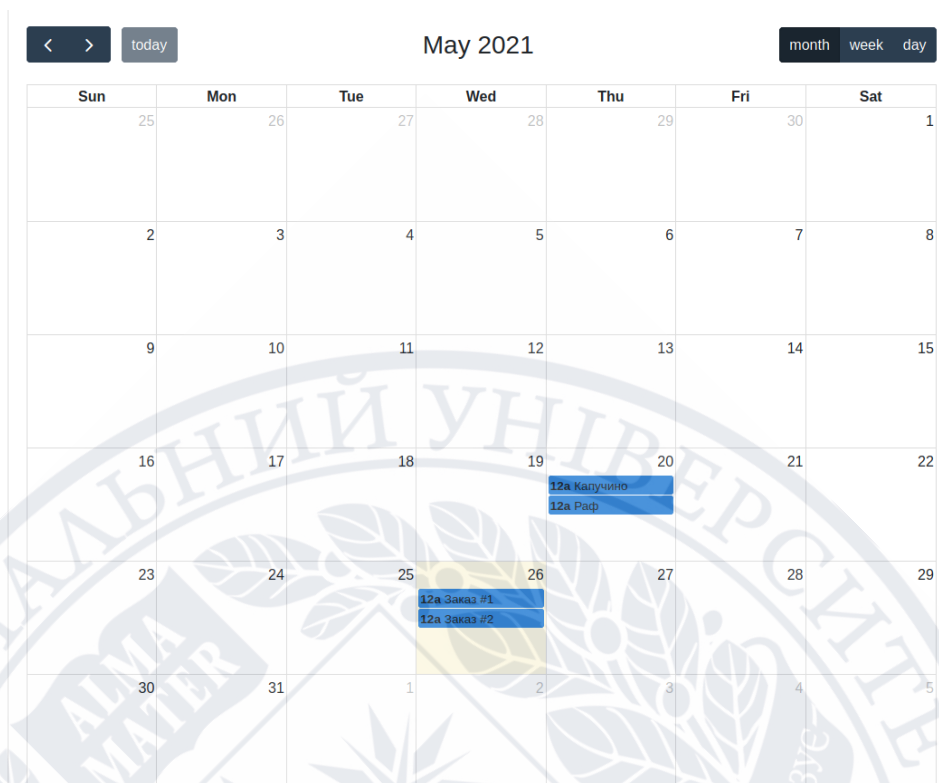


Рисунок 3.8 – Календар планування замовлень

На рис. 3.8 наведено календар планування замовлень. Календар був доданий для того, щоб замовлення, які були зроблені заздалегідь, можна внести в календар і виконати в той день, на який клієнт замовив продукт.

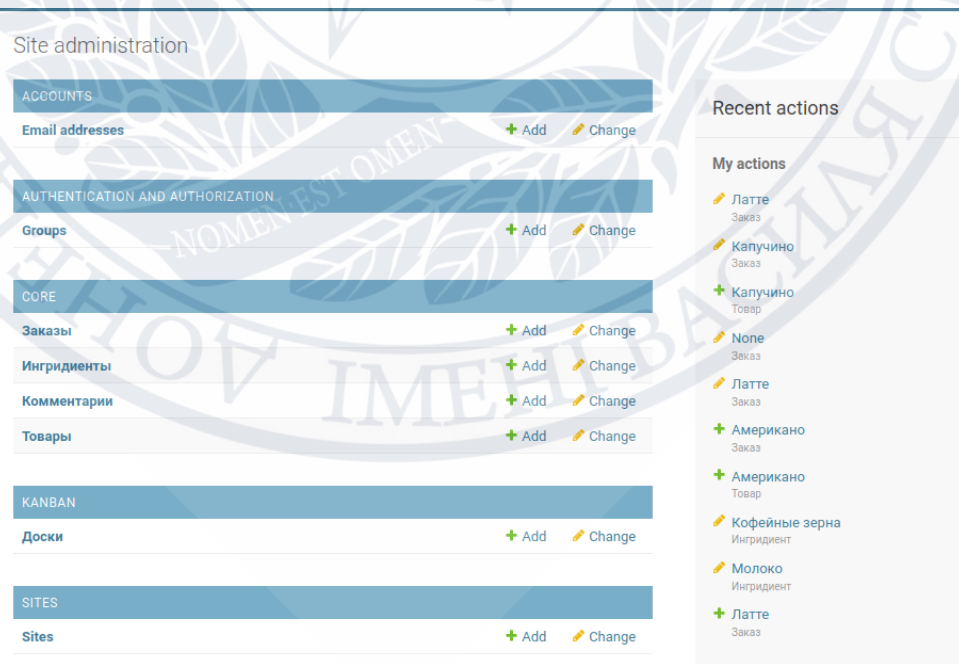


Рисунок 3.9 – Адмін-панель

На рис. 3.9 зображено Адмін-панель, яка є стандартною для фреймворка Django. Адмін-панель використовується для швидкого і зручного редагування баристою (адміністратором). Через неї формується список інгредієнтів для формування товару на складі та створення напою.

Select Товар to change ADD TOBAR +

Action: Go 0 of 3 selected

НАИМЕНОВАНИЕ ТОВАРА	GET COMPONENT LIST	GET INGREDIENTS CONSUMPTION
<input type="checkbox"/> Капучино	Кофейные зерна, Молоко	0.32
<input type="checkbox"/> Американо	Кофейные зерна	0.07
<input type="checkbox"/> Латте	Кофейные зерна, Молоко	0.32

3 Товари

Рисунок 3.10 – Склад напою в Адмін-панелі

На рис. 3.10 зображено, як виглядає склад готового напою в Адмін-панелі Django. Вказана назва самого напою, які інгредієнти використовуються і в якій кількості.

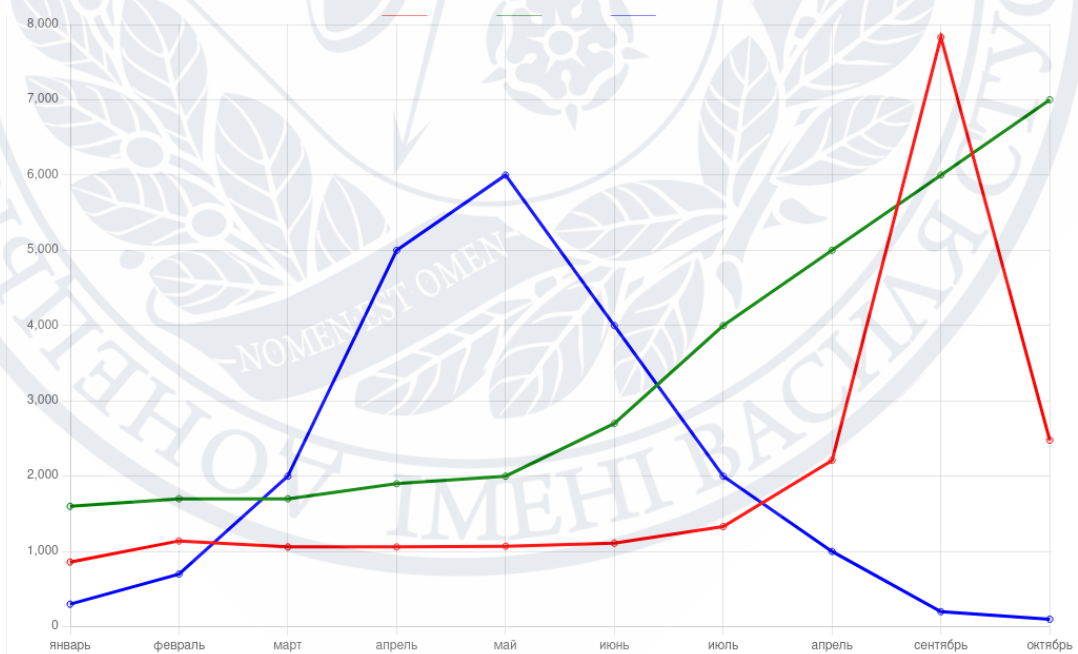


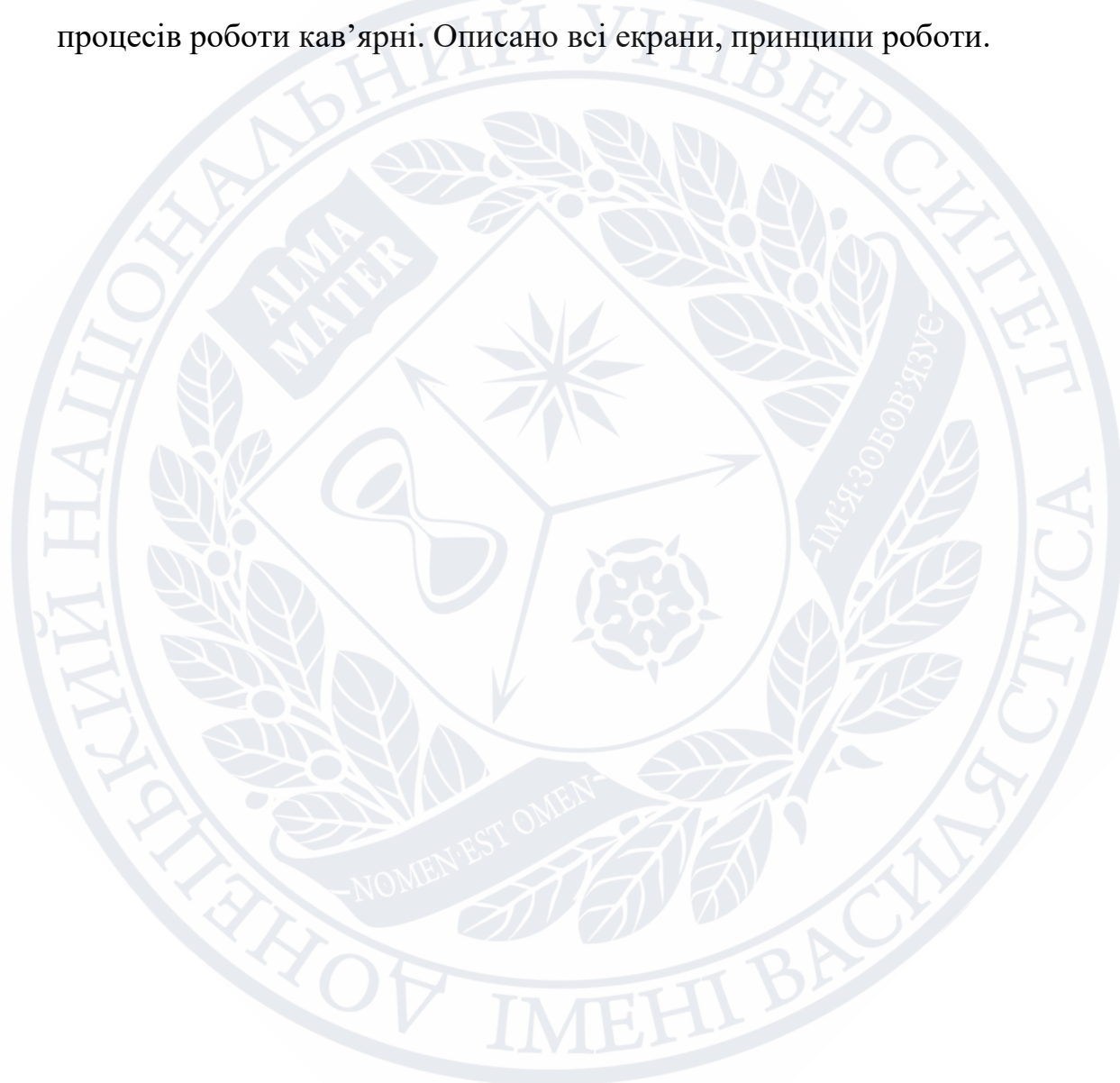
Рисунок 3.11 – Статистика замовлень помісячно

На рис. 3.11 зображено статистичний графік замовлень, які були оброблені Kanban – дошкою помісячно. На рисунку можна бачити середню кількість

замовлень для кожного напою в той чи інший місяць. З рис 3.2 були взяті напої, які використовувались для створення даного графіку. Латте – синій колір, капучіно – зелений колір і американо – червоний колір.

Висновок до розділу 3

В даному розділі було описано розроблений web-додаток для автоматизації процесів роботи кав'ярні. Описано всі екрани, принципи роботи.



ВИСНОВОК

Розробка автоматизованих систем управління є актуальною темою для роботи. Дані системи дозволяють оптимізувати процеси управління довільним підприємством. Актуальність даної роботи полягає в тому, що такий бізнес, як «Кав'ярні» є досить популярним на сьогоднішній день, але систем управління для таких підприємств дуже мало. Тому було вирішено розробити автоматизовану систему управління для обліку роботи процесів кав'ярні. Був проведений порівняльний аналіз зі схожими системами. Для порівняльного аналізу було обрано 2 web – додатки. Були підкреслені їх індивідуальні особливості.

Для досягнення мети дослідження було вивчено методику проектування web–додатків з використанням мови програмування Python та фреймворка Django.

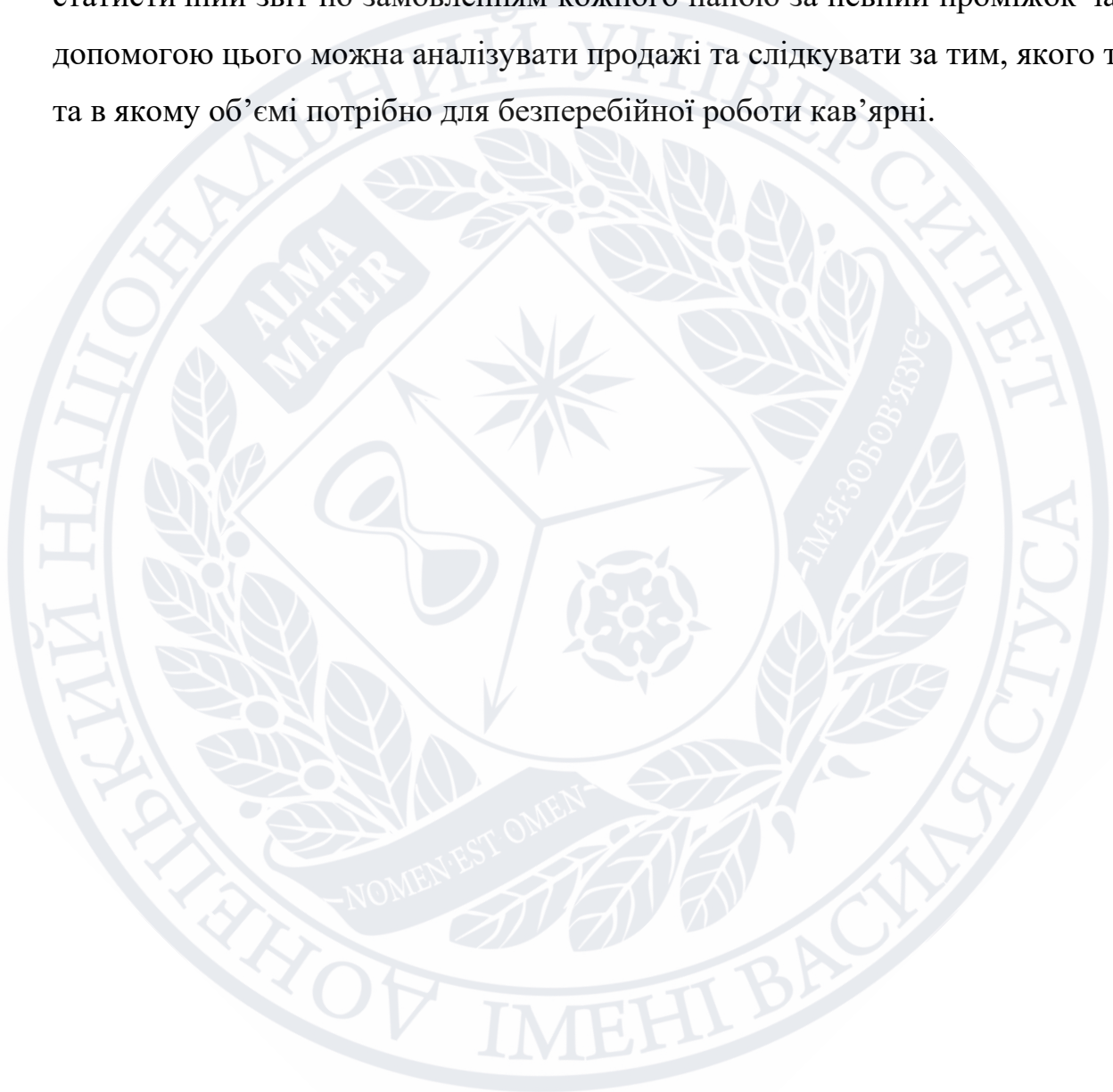
Постановка задачі полягала в розробці web–додатку для автоматизації процесів управління кав'ярні. Для розв'язання неї були окреслені наступні задачі дослідження: огляд існуючих аналогів з виявленням їх переваг та недоліків; вивчення спеціалізованого середовища розробки Visual Studio Code; розробка автоматизованої системи управління для обліку роботи процесів кав'ярні.

Були використанні такі інструменти для розробки: мова програмування Python, фреймворк Django, база даних PostgreSQL, середовище розробки Visual Studio Code, бібліотека для аналізу даних pandas, бібліотека для швидких обрахунків NumPy, фреймворк для HTML, CSS, JS – Bootstrap, для створення інтерактивних візуалізацій D3.JS, технологія звернення до серверу AJAX. Також був використаний архітектурний шаблон MVT – для проектування взаємодії елементів додатку.

Логіка web–додатку побудована на основі використання Kanban–дошки. Це дозволило всі замовлення поділити за статусами: «В обробці», «В процесі», «Виконаний», «Заблокований». Поділ за статусами дозволяє баристі слідкувати за кожним замовленням.

Система дозволяє переглядати наявну кількість товару на складі, слідкувати за терміном придатності кожного товару. В свою чергу, це дозволяє уникати простроченого товару та економить кошти на закупку.

За допомогою адмін-панелі можна налаштовувати, скільки та якого товару необхідно для приготування того чи іншого напою. Також можна переглядати статистичний звіт по замовленням кожного напою за певний проміжок часу. За допомогою цього можна аналізувати продажі та слідкувати за тим, якого товару та в якому об'ємі потрібно для безперебійної роботи кав'ярні.



СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кав'ярня як різновид спеціалізованих закладів ресторанного господарства [Електроний ресурс]. – 2018. Режим доступу до ресурсу: <http://ujae.org.ua/kav-yarnya-yak-riznovyd-spetsializovanyh-zakladiv-restorannogo-gospodarstva/>
2. Хмарна система jSolution [Електроний ресурс] – Режим доступу до ресурсу: <https://jsolutions.ua/ua/sistema-upravleniya-restoranom%203>
3. Програма для обліку в пекарні [Електроний ресурс] – Режим доступу до ресурсу: <https://skyservice.pro/uk/automation/bakery/>
4. Автоматизація кав'ярні Poster [Електроний ресурс] – 2017. Режим доступу до ресурсу: <https://joinposter.com/business/coffeeshop>
5. SmartTouch POS [Електроний ресурс] – 2016. Режим доступу до ресурсу: <https://smarttouchpos.eu/>
6. Методологія Kanban [Електроний ресурс] –2018. Режим доступу до ресурсу: https://skillbox.ru/media/management/vse_chto_nuzhno_znat_o_kanban/
- 7: Методологія Kanban [Електроний ресурс].–2018. Режим доступу до ресурсу: <https://www.atlassian.com/ru/agile/kanban/boards#:~:text=%D0%94%D0%BE%D1%81%D0%BA%D0%B0%20Kanban%20%E2%80%94%D1%8D%D1%82%D0%BE%20%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F,%D0%B8%20DevOps%20%D1%83%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BE%D1%87%D0%B8%D1%82%D1%8C%20%D0%BF%D0%BE%D0%B2%D1%81%D0%B5%D0%B4%D0%BD%D0%B5%D0%B2%D0%BD%D1%83%D1%8E%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%83.>
8. Що таке Kanban [Електроний ресурс].–.Режим доступу до ресурсу: <https://www.atlassian.com/ru/agile/kanban>

9. Swaroop C.H. A byte of Python
10. Реалізація MVT паттерна [Електроний ресурс]. -2019. Режим доступу до ресурсу: <https://ru.hexlet.io/blog/posts/pochemu-django-luchshiy-freymvork-dlya-razrabotki-saytov>
11. Що таке Django [Електроний ресурс]. –2019. Режим доступу до ресурсу: <https://tutorial.djangogirls.org/ru/django/>
12. Django введення [Електроний ресурс]. –2021. Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction>
13. Чому Django – кращий фреймворк для розробки сайтів [Електроний ресурс]. - 2019. Режим доступу до ресурсу: <https://ru.hexlet.io/blog/posts/pochemu-django-luchshiy-freymvork-dlya-razrabotki-saytov>
14. PostgreSQL [Електроний ресурс]. – 2019. Режим доступу до ресурсу: <https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-postgresql/postgresql-kratkoe-rukovodstvo>
15. Чим PostgreSQL кращий за другі SQL бази даних [Електроний ресурс]. –2016. Режим доступу до ресурсу: <https://habr.com/ru/post/282764/>
16. PostgreSQL – об'єктно-реляційна система управління баз даних [Електроний ресурс] – Режим доступу до ресурсу: <https://web-creator.ru/articles/postgresql>
17. Visual studio code [Електроний ресурс]. – 2021. Режим доступу до ресурсу: <https://htmlacademy.ru/blog/boost/tools/vscode>
18. Python Pandas [Електроний ресурс]. – 2018. Режим доступу до ресурсу: <https://coderlessons.com/tutorials/python-technologies/vyuchit-python-panda/python-pandas-kratkoe-rukovodstvo>
19. NumPY [Електроний ресурс]. – 2015. Режим доступу до ресурсу: <https://pythonworld.ru/numpy/1.html>
20. NumPY в Python [Електроний ресурс]. – 2018. Режим доступу до ресурсу: <https://habr.com/ru/post/352678/>
21. NumPY [Електроний ресурс]. – 2020. Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/NumPy>

22. Що таке Bootstrap і навіщо він потрібен [Електроний ресурс]. – 2021. Режим доступу до ресурсу: <https://itchief.ru/bootstrap/introduction>

23. D3.JS [Електроний ресурс]. – 2018. Режим доступу до ресурсу: <https://coderlessons.com/tutorials/veb-razrabotka/uznaite-d3js/d3-js-kratkoe-rukovodstvo>

24: Введення в Аjaх [Електроний ресурс]. –2019. Режим доступу до ресурсу: <https://learn.javascript.ru/ajax-intro>

25:Аjaх для новачків [Електроний ресурс]. –2007. Режим доступу до ресурсу: <https://habr.com/ru/post/14246/>



Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

Сеник Іван Олександрович

Прізвище, ім'я, по батькові

Інформаційних і прикладних технологій

Факультет

122 «Комп'ютерні науки»

Шифр і назва спеціальності

«Сучасні інформаційні технології та програмування»

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «Розробка автоматизованої системи управління для кав'ярні» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис