

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка плагіна “Часті питання” для інтернет-магазину
на базі CMS PrestaShop

Виконав: студент IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Столяр А.В.

(прізвище та ініціали)

Керівник

(підпис)

Гладько Ю.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Стоянов Ю.М.

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«___» червня 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Столяр Альоні Володимирівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка плагіна "Часті питання" для інтернет-магазину на базі CMS PrestaShop.

Керівник роботи Гладько Юрій Богданович, кандидат технічних наук, доцент кафедри ММ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-470

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Літературні та інтернет джерела щодо використання плагінів, розробки плагінів на базі CMS PrestaShop.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1.1 Актуальність використання плагінів у інтернет-магазинах. 1.2 Огляд системи управління контентом CMS. 1.3 Аналіз CMS PrestaShop. 2.1 Вимоги до плагіна "Часті питання". 2.2 Вибір мови програмування для розробки плагіна. 2.3 Вибір середовища розробки. 2.4 Взаємодія з ядром "PrestaShop". 2.5 Визначення акторів та діаграм варіантів використання. 2.6 Формування архітектури плагіна "Часті питання". 2.7 Проєктування бази даних. 2.8 Проєктування структури функціональних компонентів. 2.9 Файлова структура плагіна. 3.1 Розробка плагіна "Часті питання". 3.2 Огляд функціоналу CMS платформи PrestaShop. 3.3 Впровадження плагінів Літературні та інтернет джерела щодо веб-технологій розроблення плагіна в інтернет-магазин. 3.4 Приклади використання плагіна "Часті питання".

4. Безпека життєдіяльності, основи охорони праці. 4.1 Роль центральної нервової системи в трудовій діяльності людини. 4.2 Перша допомога людині, яка уражена електричним струмом. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний слайд. 2. Актуальність використання плагінів. 3. Огляд системи управління контентом (CMS). 4. Аналіз CMS PrestaShop. 5. Вимоги до плагіна "Часті питання".

6. Взаємодія з ядром "PrestaShop". 7. Діаграми варіантів використання. 8. Формування архітектури плагіна. 9. Модель сутностей бази даних. 10. Огляд сторінки "Dashboard" CMS PrestaShop. 11. Сторінка "Створення питання та відповіді". 12. Сторінка "Створення категорій". 13. Сторінка плагіна "Часті питання". 14. Форма "Ask a question". 15. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В. С., кандидат технічних наук, доцент кафедри МТ		

7. Дата видачі завдання _____ 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел про розробку плагінів для CMS	31.01.2024-03.02.2024	
3.	Опрацювання джерел про розробку плагінів для Інтернет-магазинів на базі CMS PrestaShop	04.02.2024-06.02.2024	
4.	Розробка плагіна «Часті питання»	07.02.2024-11.02.2024	
5.	Оформлення розділу «Теоретичні аспекти плагіна «Часті питання»»	03.06.2024-05.06.2024	
6.	Оформлення розділу «Проектування плагіна «Часті питання»»	06.06.2024-08.06.2024	
7.	Оформлення розділу «Практична реалізація плагіна «Часті питання»»	09.06.2024-11.06.2024	
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	
9.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	
10.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	
11.	Нормоконтроль	18.06.2024-19.06.2024	
12.	Перевірка на плагіат	10.06.2024	
13.	Попередній захист кваліфікаційної роботи	21.06.2024	
14.	Захист кваліфікаційної роботи	27.06.2024	

Студент

_____ (підпис)

Столяр А.В.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Гладь Ю.Б.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка плагіна «Часті питання» для інтернет-магазину на базі CMS PrestaShop// Кваліфікаційна робота освітнього рівня «Бакалавр»// Столяр Альона Володимирівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2024 // С. 69, рис. – 23, табл. – 8, додат. – 3 , бібліогр. – 36.

Ключові слова: плагін, CMS, PrestaShop, база даних, PHP, HTML, CSS, JavaScript, SQL .

Кваліфікаційна робота присвячена розробці плагіна «Часті питання» для інтернет-магазину на базі CMS PrestaShop.

В першому розділі кваліфікаційної роботи описано теоретичні аспекти плагіна «Часті питання». Проаналізовано актуальність використання плагінів в інтернет-магазинах та розглянуто особливості розробки плагінів для CMS PrestaShop.

В другому розділі кваліфікаційної роботи досліджено взаємодію з ядром PrestaShop та визначено основні вимоги до плагіна. До того ж, проведено проектування бази даних та сформовано архітектуру плагіна «Часті питання».

В третьому розділі кваліфікаційної роботи описано практичну реалізацію плагіна «Часті питання» для інтернет-магазинів на базі CMS PrestaShop. Проведено встановлення плагіна на CMS PrestaShop та огляд функціоналу розробленого плагіна. Крім того, описано інструкцію щодо встановлення плагіна та наведено основні приклади його використання.

ANNOTATION

Development of the plugin «Frequently Asked Questions» for an online store based on CMS PrestaShop // Qualification work of the educational level "Bachelor" // Stolyar Aliona // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2024 // P. 69, fig. - 23, tabl. - 8, annexes. – 3, references - 36.

Keywords: plugin, CMS, PrestaShop, database, PHP, HTML, CSS, JavaScript, SQL.

The qualification work is devoted to the development of the plugin «Frequently Asked Questions» for an online store based on CMS PrestaShop.

The first chapter of the qualification work describes the theoretical aspects of the “Frequently Asked Questions” plugin. The relevance of using plugins in online stores is analyzed and the peculiarities of developing plugins for CMS PrestaShop are considered.

The second section of the qualification work examines the interaction with the PrestaShop core and identifies the basic requirements for the plugin. In addition, the database was designed and the architecture of the plugin «Frequently Asked Questions» was formed.

The third section of the qualification work describes the practical implementation of the plugin «Frequently Asked Questions» for online stores based on CMS PrestaShop. The plugin was installed on CMS PrestaShop and the functionality of the developed plugin was reviewed. In addition, the instructions for installing the plugin are described and the main examples of its use are given.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

AJAX (англ. Asynchronous JavaScript and XML) – асинхронний JavaScript і XML.

CMS (англ. Content Management System) – система керування вмістом.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

HTML (англ. Hyper Text Markup Language) – мова розмітки гіпертексту.

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки для розробки програмного забезпечення.

SEO (англ. Search Engine Optimization) – пошукова оптимізація сайту.

SQL (англ. Structured Query Language) – мова структурованих запитів, призначена для взаємодії з базами даних.

PHP (англ. HyperText Preprocessor) – скриптова мова програмування.

БД – база даних.

СКБД – система керування (управління) базами даних.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ПЛАГІНА “ЧАСТІ ПИТАННЯ”	11
1.1 Актуальність використання плагінів у інтернет-магазинах	11
1.2 Огляд системи управління контентом (CMS)	12
1.3 Аналіз CMS PrestaShop	14
1.3.1 Основні функції та можливості PrestaShop	15
1.3.2 Особливості розробки плагінів для PrestaShop	16
1.4 Висновок до першого розділу	18
РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАГІНА “ЧАСТІ ПИТАННЯ”	20
2.1 Вимоги до плагіна “Часті питання”	20
2.2 Вибір мови програмування для розробки плагіна	21
2.3 Вибір середовища розробки	23
2.4 Взаємодія з ядром “PrestaShop”	24
2.5 Визначення акторів та діаграми варіантів використання	25
2.6 Формування архітектури плагіна “Часті питання”	27
2.7 Проєктування бази даних	29
2.8 Проєктування структури функціональних компонентів	35
2.9 Файлова структура плагіна “Часті питання”	37
2.10 Висновок до другого розділу	41
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПЛАГІНА “ЧАСТІ ПИТАННЯ”	42
3.1 Розробка плагіна “Часті питання”	42
3.1.1 Підготовка середовища розробки	42
3.1.2 Реалізація основних функцій плагіна	43
3.2 Огляд функціоналу CMS платформи PrestaShop	44
3.3 Впровадження плагіна в інтернет-магазинах	47
3.3.1 Інструкція зі встановлення плагіна	48
3.3.2 Налаштування плагіна та огляд основних можливостей	49
3.4 Приклади використання плагіна “Часті питання”	54

3.4.1 Приклад використання плагіна в інтернет-магазині	54
3.4.2 Приклад використання форми для задання власного питання	57
3.5 Висновок до третього розділу	59
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	60
4.1 Роль центральної нервової системи в трудовій діяльності людини ..	60
4.2 Перша допомога людині, яка уражена електричним струмом	62
ВИСНОВКИ	65
ПЕРЕЛІК ДЖЕРЕЛ	66
ДОДАТКИ	

ВСТУП

Актуальність теми. У зв'язку із стрімким розвитком електронної комерції інтернет-магазини стають все більш поширеними для торгівлі послуг та товарів. Зростання кількості інтернет-магазинів в результаті веде до підвищення конкуренції, що змушує власників бізнесу шукати нові можливості та способи для залучення клієнтів та підвищення їхньої задоволеності. Тому, одним із ефективних інструментів для досягнення цих цілей є використання плагінів, які є допоміжним функціоналом у розширенні можливостей для інтернет-магазинів. Зокрема, плагін “Часті питання” дозволяє спрощено відповідати на типові, найбільш розповсюджені запити користувачів, що допомагає мінімізувати навантаження на службу підтримки та збільшити якість рівня обслуговування клієнтів.

CMS PrestaShop є однією з найбільш популярних платформ для створення інтернет-магазинів завдяки своїй гнучкості, широкому функціоналу та активній спільноті розробників, які допомагають розширити функціонал та зробити використання платформи ще зручнішим. Щодо розробки плагіна “Часті питання” на базі платформи PrestaShop, то він дозволить підвищити ефективність управління інтернет-магазином та значно покращити користувацький досвід, створивши добре структуровану, легко керовану сторінку з усією інформацією, яка може знадобитися користувачам.

Мета і задачі дослідження. Метою розробки плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop полягає в забезпеченні ефективної та швидкою підтримки користувачів на їх запитання та сприянню зменшення навантаження на службу підтримки.

Для досягнення зазначеної мети було сформовано ряд основних завдань:

- Проведення дослідження для визначення основних вимог щодо плагіна.
- Розробка архітектури плагіна.
- Реалізація плагіна та його інтеграція з інтернет-магазином на базі PrestaShop.

- Тестування плагіна та оптимізація його роботи.
- Підготовка інструкції із встановлення та використання плагіна.

Об'єкт дослідження. Об'єктом дослідження в даній роботі є процес розробки плагіна “Часті питання” на базі CMS PrestaShop та його вплив на підвищення ефективності управління інтернет-магазином.

Предмет дослідження. Предметом дослідження в даній роботі є технологія розробки плагіна “Часті питання”, який спрямований на покращення доступності користувачам відповіді на їх запитання. Крім того, дослідження що аналіз вимог до функціональності, вибору технологій, проектування, реалізацію та тестування даного плагіна.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ПЛАГІНА “ЧАСТІ ПИТАННЯ”

1.1 Актуальність використання плагінів у інтернет-магазинах

У сучасному динамічному світі інтернет-магазини займають вагоме місце в електронній комерції, стаючи невід'ємною складовою багатьох бізнес-процесів. Інтернет-магазини функціонують у надзвичайно швидкому та конкурентному середовищі, де успіх значною мірою залежить від здатності швидко адаптуватися до змін ринку та задовольняти потреби клієнтів. Впровадження плагінів стає все більш поширеною практикою і незамінним інструментом для досягнення гнучкості, масштабованості та інновацій у розвитку електронної комерції [1].

Плагіни слугують додатковими програмними модулями, які дозволяють значно розширити функціональні можливості систем управління контентом (CMS) без необхідності глибокого втручання в основний код системи, що веде за собою низку переваг. Розробка плагінів впершу чергу забезпечує розширення функціональності магазину, надаючи можливість додавати нові функції та інтегрувати нові технології до вже існуючого функціоналу магазину [2]. Це може включати різні інструменти для оптимізації SEO, імпорту та експорту товарів, обробки замовлень або ж взаємодії з клієнтами. Використання готових рішень у вигляді плагінів значно знижує витрати на розробку потреб бізнесів. Власники інтернет-магазинів можуть скористатися вже раніше створеними плагінами, які пройшли тестування та мають хорошу репутацію серед користувачів, що забезпечує високу надійність та якість їхньої роботи [3]. До того ж, використання даних плагінів значно мінімізує процес налаштування та керування інтернет-магазином, що дає змогу власникам-бізнесу зосередитися на більш важливій діяльності.

Для популярних CMS використання таких розширень як плагіни в інтернет-магазинах існує велика спільнота розробників, яка на постійній основі розробляє та підтримує їх, що забезпечує широкий спектр рішень та здатність

отримати якомога швидшу підтримку вразі виникнення проблем. До того ж, більшість плагінів мають простий та досить зрозумілий процес встановлення та налаштування, що дозволяє навіть користувачам без глибоких технічних знань швидко інтегрувати нові функції у свій інтернет-магазин. Більш того, плагіни регулярно оновлюються для забезпечення сумісності з новими версіями CMS та виправлення виявлених недоліків, що гарантує їхню актуальність та надійність [4]. Такі розширення є модульними складовими, що дозволяє легко додавати або видаляти функціонал плагіна без ризику порушити роботу сайту, що й надає високу гнучкість в управлінні усім функціоналом магазину.

Таким чином, плагіни для інтернет-магазинів на базі системи управління контентом CMS займають важливу роль у розвитку, вдосконаленні та функціонуванні магазину, забезпечуючи необхідні інструменти для оптимізації та автоматизації бізнес-процесів.

1.2 Огляд системи управління контентом (CMS)

Системи управління контентом (CMS) стали незамінними інструментами в управлінні цифровим контентом, особливо для веб-сайтів та інтернет-магазинів. Ці системи забезпечують надійну основу для створення, управління та модифікації цифрового контенту, що робить їх важливими для бізнесу будь-якого розміру. Тобто, CMS – це програмне забезпечення, яке слугує для організації веб-сайтів та надає можливість для створення, керування, редагування або ж публікування контенту без необхідності технічних знань, що й робить дані системи незамінними для бізнесів, які прагнуть бути на рівні сучасних вимог для своїх веб-ресурсів і швидко адаптовуватися до змін ринку. Таке програмне забезпечення дозволяє користувачам створювати та керувати веб-сайтом без необхідності кодувати його з нуля або ж взагалі не кодувати, що і свідчить про суттєву користь та перевагу [5]. Користувач може налаштовувати дизайн, функціональність сайту на CMS платформі лише завдяки використанню плагінів.

В основі будь-якої CMS лежить її здатність полегшувати створення та редагування контенту. Це стосується не лише тексту, але й мультимедійної частини. Власники інтернет-магазинів здатні використовувати CMS для легкого оновлення описів товарів, завантаження нових зображень товарів та управління запасами без будь-яких навичок програмування. Ця можливість і є однією з переваг CMS – управління структурою сайту, що дає змогу створювати сторінки, категорії та їх організувати та забезпечувати зручну та структурну навігацію магазину [6]. Це в свою чергу забезпечує легкий доступ до необхідної інформації, що й і є ще однією важливою функцією CMS-платформ – здатність керувати структурою сайту.

Добре організована структура покращує користувацький досвід, полегшуючи відвідувачам пошук інформації, яку вони шукають. Слід також зазначити, що важливим критерієм є безпека інтернет-магазину та його управлінням, безпечність транзакцій та захист від несанкціонованого доступу. Системи управління контентом забезпечує права доступу до магазину та управління ролями, які дозволяють надавати права доступу та контролювати дії, які можуть виконувати користувачі на сайті. В свою чергу, регулярні оновлення від розробників CMS ще більше підвищують безпеку, усуваючи вразливості та покращуючи стабільність системи.

Що стосується глобальності використання CMS платформ, то більшість даних платформ пропонують підтримку багатомовного контенту, що має вирішальне значення для компаній, які працюють у кількох регіонах або обслуговують різні мовні аудиторії. Ця функція дозволяє веб-сайтам представляти контент різними мовами, задовольняючи потреби ширшої аудиторії та підвищуючи залученість користувачів. Багатомовна підтримка не лише допомагає вийти на глобальний ринок, але й покращує пошукову оптимізацію SEO різними мовами, підвищуючи видимість і доступність веб-сайту.

Аналізуючи поширеність використання плагінів на даних платформах, то ця можливість є доволі великим плюсом у роботі CMS платформ, так як

користувачі можуть вибирати з широкого спектру плагінів, ті, які їм вкрай необхідні для оптимізації та покращення роботи всього магазину та їх бізнесу [7].

Використовуючи готові шаблони та плагіни, компанії можуть запускати веб-сайти та керувати ними швидше і з меншими витратами ресурсів, ніж при створенні сайту з нуля. Така ефективність особливо вигідна для малих і середніх підприємств, які можуть не мати бюджету на масштабну індивідуальну розробку. Крім того, масштабованість платформ CMS означає, що вони можуть рости разом з бізнесом, пристосовуючись до зростаючого трафіку і більш складної функціональності в міру необхідності [8].

Отже, CMS є важливими інструментами для створення та управління сучасними інтернет-магазинами. Їхні можливості у створенні контенту, управлінні структурою веб-сайту, інтеграції зі сторонніми розробниками, безпеці, багатомовній підтримці, плагінам роблять їх надзвичайно необхідними для бізнесу, який прагне вдосконалюватися.

1.3 Аналіз CMS PrestaShop

PrestaShop – це універсальна та надійна система управління контентом (CMS), розроблена спеціально для додатків електронної комерції. Як зазначалося в попередньому розділі, важливість платформ CMS полягає в їхній здатності оптимізувати управління цифровим контентом і завдяки модульній архітектурі забезпечувати широкі можливості кастомізації та масштабування [9].

PrestaShop виділяється на переповненому ринку CMS завдяки широкому спектру функцій, пристосованих до потреб інтернет-магазинів, і потужній архітектурі плагінів, яка полегшує розробку кастомних функцій, таких як плагін “Часті питання”.

Поєднання модульності, комплексного управління продуктами, гнучкості дизайну, підтримки, надійної безпеки та оптимізації продуктивності робить PrestaShop однією з кращих CMS-платформ для електронної комерції. Ці функції

не лише задовольняють різноманітні потреби інтернет-магазинів, але й створюють міцну основу для розробки плагінів.

1.3.1 Основні функції та можливості PrestaShop

Основний функціонал PrestaShop призначений для підтримки повного життєвого циклу інтернет-магазину, від створення списку товарів і управління запасами до обробки замовлень і обслуговування клієнтів. Однією з головних переваг PrestaShop є гнучкість в управлінні широким спектром продуктів, включаючи складні варіації та комбінації продуктів. Вона підтримує кілька валют і мов, що важливо для компаній, які працюють у різних географічних регіонах. Платформа також включає розширені правила ціноутворення, рекламні інструменти та функції SEO-оптимізації, які мають вирішальне значення для підвищення видимості та конкурентоспроможності інтернет-магазину [10].

Можливості бек-офісу PrestaShop не менш вражають, надаючи адміністраторам магазинів комплексні інструменти для управління замовленнями, відстеження продажів і аналізу ефективності за допомогою детальних звітів та аналітики. Ці функції дозволяють продавцям приймати рішення на основі даних та оптимізувати свою діяльність. Крім того, PrestaShop легко інтегрується з різними платіжними шлюзами та перевізниками, забезпечуючи безперебійний і безпечний процес транзакцій для клієнтів.

Однією з визначних особливостей PrestaShop є його модульна архітектура, яка дозволяє користувачам розширювати функціональність своїх магазинів за допомогою широкого спектру плагінів. Ця модульність є основою адаптивності платформи, що дозволяє розробникам створювати індивідуальні рішення, які відповідають конкретним потребам бізнесу. Маркетплейс PrestaShop Addons пропонує тисячі модулів, що охоплюють різноманітні функціональні можливості, від інструментів маркетингу та SEO до розширеного управління продуктами та покращення обслуговування клієнтів. Така розширюваність дозволяє впроваджувати унікальні функції, такі як складний розділ поширених

запитань і відповідей, які можуть покращити взаємодію з клієнтами та зменшити навантаження на службу підтримки.

Що стосується взаємодії з користувачем, PrestaShop пропонує інтуїтивно зрозумілий інтерфейс, який легко налаштовується. Система шаблонів забезпечує широку гнучкість дизайну, гарантуючи, що бізнес може створити унікальну та цікаву користувачеві вітрину магазину. Платформа підтримує безліч тем і макетів, які можна налаштувати відповідно до фірмового стилю магазину. Ця гнучкість дизайну доповнюється потужними можливостями SEO PrestaShop, які допомагають підвищити рейтинг в пошукових системах і залучити органічний трафік до магазину [11].

Підсумовуючи можна зазначити, що PrestaShop пропонує комплексний набір функцій і досить гнучку архітектуру, що робить його потужним вибором для бізнесів електронної комерції. Широкі функціональні можливості даної платформи CMS в поєднанні з модульними розширеннями забезпечують значну масштабованість та оптимізацію, що надає змогу розробникам створювати індивідуальні рішення під будь-які потреби.

1.3.2 Особливості розробки плагінів для PrestaShop

Однією з ключових особливостей розробки плагінів для PrestaShop є використання модульної архітектури. Плагіни в PrestaShop — це незалежні компоненти, які можуть бути додані до системи без необхідності втручання в основний код. Це дозволяє зберегти цілісність та стабільність основної платформи, водночас додаючи нові функції. Розробники можуть створювати модулі для різних цілей: імпорту/експорту продуктів або ж замовлень, інтеграції платіжних систем, маркетингових інструментів, аналітики тощо.

Розробка плагінів для PrestaShop вимагає глибокого розуміння його архітектури. Платформа PrestaShop базується на мові програмування PHP, JavaScript, HTML, CSS та базі даних MySQL. PHP використовується для написання серверної логіки плагіна, обробки даних та взаємодії з базою даних.

Щодо бази даних MySQL, то вибір був очевидним, так як це стандартна база даних для CMS PrestaShop та є повністю інтегрованою з ядром PrestaShop та дозволить уникнути проблем із сумісністю. HTML та CSS використовуються для створення інтерфейсу користувача, а JavaScript — для реалізації динамічної взаємодії та анімації. Тому розробники повинні дотримуватися стандартів і рекомендацій PrestaShop щодо кодування, щоб забезпечити сумісність і продуктивність [12].

Щодо сумісності модулів з платформою PrestaShop, важливо враховувати версії ядра PrestaShop та сумісність модулів з ними. Оскільки PrestaShop регулярно оновлюється, розробники повинні періодично оновлювати свої модулі, щоб забезпечити їхню сумісність з останніми версіями платформи. Деякі модулі можуть також мати вимоги до певних версій PHP, MySQL та інших залежностей, які також слід враховувати при виборі модулів для встановлення в магазині.

Створення плагіна включає в себе кілька ключових кроків. Спершу необхідно визначити функціональні вимоги до модуля та його основні завдання. Після цього розробляється структура модуля, яка включає необхідні файли та каталоги, такі як контролери, моделі, шаблони та мовні файли. Використання стандартів кодування PrestaShop та дотримання його архітектурних принципів є критично важливими для забезпечення сумісності та надійності модуля. Розробка починається з визначення та налаштування його параметрів, в основному PHP-файлів модуля. Наступним кроком зазвичай є створення необхідних шаблонів і JavaScript-файлів для управління користувацьким інтерфейсом і взаємодією. Система хуків PrestaShop є ключовою в цьому процесі, оскільки вона дозволяє модулям взаємодіяти з ядром системи та іншими модулями без зміни основного коду. Це особливо корисно для розробки плагінів, таких як модуль поширених запитань, які можуть потребувати інтеграції з різними частинами магазину, наприклад, зі сторінками продуктів і процесом оформлення замовлення [13].

Отже, розуміння вимог до середовища розробки CMS PrestaShop, функціональні вимоги до плагіна та його структури відіграє вирішальне значення для розробки та використання високоякісних плагінів, які покращують загальну функціональність та автоматизацію інтернет-магазинів.

1.4 Висновок до першого розділу

Отже, перший розділ цього дослідження містить огляд теоретичних аспектів розробки плагіна для інтернет-магазину на базі CMS PrestaShop. Аналіз підкреслює важливість плагінів для підвищення функціональності та продуктивності платформ електронної комерції, особливо динамічність ринку електронної комерції, де адаптивність та ефективність мають першочергові значення.

Вивчення систем управління контентом висвітлює їхню ключову роль у полегшенні управління веб-сайтами та створенні контенту. Платформи CMS, такі як PrestaShop, пропонують багато функцій і можливостей, пристосованих до потреб інтернет-магазинів, що дозволяє їм ефективно управляти продуктами, замовленнями, платежами та взаємодією з клієнтами.

Щодо аналізу особливостей розробки плагінів для PrestaShop, то варто зазначити, що розробники повинні враховувати багато вагомих критеріїв, щоб забезпечити ефективність та сумісність своїх плагінів з платформою PrestaShop. Спершу, це модульна архітектура PrestaShop, яка є фундаментальною особливістю та значно впливає на розробку плагінів. Крім того, необхідне глибоке розуміння технологій, що лежать в основі PrestaShop, таких як PHP, JavaScript, HTML та CSS, які є невід'ємною частиною розробки плагінів. Дані технології становлять основу архітектури PrestaShop і використовуються для реалізації функціональності плагінів. Дотримання стандартів кодування та архітектурних принципів PrestaShop забезпечує сумісність та продуктивність, дозволяючи розробникам створювати високоякісні плагіни, які відповідають основним вимогам та стандартам платформи.

Підсумовуючи, можна сказати, що теоретичні аспекти, зроблені в цьому розділі, закладають основу для наступних етапів розробки плагінів. Висвітлюючи теоретичні основи платформ CMS, досліджуючи можливості PrestaShop і заглиблюючись у тонкощі розробки плагінів, цей розділ надає повне розуміння вимог для проектування та практичної реалізації плагіна.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАГІНА “ЧАСТІ ПИТАННЯ”

2.1 Вимоги до плагіна “Часті питання”

При проєктуванні даного плагіна необхідно врахувати усі вимоги, які необхідні для розробки. Для того, щоб забезпечити усі вимоги плагіна для інтеграції та його використання, було виділено основні із них:

- функціональні вимоги;
- вимоги графічного інтерфейсу;
- вимоги користувачів.

До функціональних вимог належать: можливість створення та редагування запитань, сумісність з CMS платформою PrestaShop, розподіл та фільтрування запитань щодо категорій та продукту. Адміністратор має мати змогу створювати запитання та редагувати вже створені раніше запитання та відповіді і також видаляти неактуальну інформацію щодо запитань. Даний плагін має забезпечувати сумісність з CMS PrestaShop згідно її вимог та версії платформи [14]. Плагін має надавати можливість адміністратору обирати та додавати запитання відповідно до актуальності та відповідності продукту та категоріям. Питання, які стосуються певної категорії, групувати по темах, мають відображатися на стороні користувача в конкретній категорії. Має бути додана властивість щодо додавання запитань та відповідей до конкретного продукту, щоб адміністратор міг на сторінці продукту додавати конкретне запитання, яке допоможе користувачам швидше знайти відповідати на свої запитання, що надасть змогу для зручної навігації користувачів.

До вимог користувачів належать: вимоги для адміністраторів, для клієнтів та для розробників. Для адміністраторів має надаватися можливість зручного інтерфейсу задля ефективного управління плагіном, а саме додаванням запитань та відповідей на них. Також плагін повинен забезпечувати додаткові налаштування, такі як вибір поширення запитань по категоріях та продуктах, зміна стилів та відображення вкладки частих питань. Розробка плагіна має мати

інтуїтивну та просту навігацію на CMS платформі PrestaShop. Плагін має надавати легкий доступ до запитань, на які клієнти шукають відповіді. Основною важливою вимогою до розробника плагіна “Часті питання” є те, що плагін має бути сумісний із платформою CMS PrestaShop та забезпечити усі потреби адміністратора магазину та користувачів, які будуть використовувати даний плагін.

Останньою складовою з усіх перелічених вимог є вимоги до графічного інтерфейсу, серед яких мають бути такі: зручність та доступність у використанні, адаптивність інтерфейсу плагіна та дизайн інтерфейсу. Найголовнішою вимогою інтерфейсу і є те, щоб плагін був зручним у використанні як для користувача так і для адміністратора інтернет-магазину. Так як клієнти впершу чергу і звертають увагу на зручність, доступність та легкість у використанні. Це є дуже важливо, так як здебільшого більшість власників інтернет-магазинів можуть не мати технічних навичок, щоб зрозуміти складний процес налаштування плагіна, встановлення та користування. Користувачі повинні мати можливість користуватися плагіном як на комп'ютерах так і на мобільних пристроях, що є досить зручним. Це надасть змогу більшій кількості користувачам використовувати цей плагін в незалежності від пристрою. Що стосується дизайну, то він має звертати увагу клієнта, до заохочення у використанні. Має бути сучасним, стильним та візуально приваблювати користувача.

Під час аналізу основних аспектів та вимог щодо плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop було виділено головні та найбільш значущі та важливі, які стосуються функціональності, інтерфейсу, вимог до користувачів та розробників.

2.2 Вибір мови програмування для розробки плагіна

Для розробки плагіна “Часті питання” для інтернет-магазину на платформі PrestaShop вибір мови, на якій буде розроблятися плагін є критично важливим вибором, який впливатиме на продуктивність, ефективність в роботі та

використанні плагіна [15]. За для сумісності із CMS PrestaShop необхідно врахувати архітектуру PrestaShop і тому, мова програмування, яка використовується для розробки плагінів є PHP. Також, для забезпечення повного функціоналу плагіна потрібно додати використання JavaScript, HTML та CSS.

Сама PrestaShop побудована на PHP, широко використовуваній мові сценаріїв з відкритим вихідним кодом, яка особливо добре підходить для веб-розробки і може бути вбудована в HTML. Інтеграція PHP з PrestaShop є безшовною, оскільки вся платформа використовує PHP для серверної логіки [16]. Дана сумісність спрощує процес розробки, оскільки немає необхідності в складній взаємодії між різними мовами програмування, так як PHP надає можливість взаємодіяти із ядром PrestaShop. При використанні PHP, розробники можуть безпосередньо взаємодіяти з основними функціями та хуками PrestaShop, що й забезпечує ефективну роботу плагіна для платформи.

Крім того, PHP забезпечує високу продуктивність, що є досить важливим для інтернет-магазинів на платформі PrestaShop, які мають великий обсяг даних та велике навантаження. В свою чергу це дуже важливо для плагіна “Часті питання”, який повинен миттєво завантажуватися, щоб надавати негайні відповіді на запити клієнтів, тим самим покращуючи їхній досвід покупок. Також мова PHP відома своєю надійністю та гнучкістю, які мають суттєве значення для розробки плагіна “Часті питання”.

Ще також слід зазначити такий важливий аспект для розробки плагіна використовуючи мову програмування PHP, як безпека. PHP надає надійні засоби захисту [17]. Враховуючи, що плагін “Часті питання” буде обробляти дані, введені користувачем, які є конфіденційними, то використання здатності безпеки PHP має важливе значення для захисту як даних користувача, так і усього інтернет-магазину.

Окрім PHP, розробка плагіна для платформи PrestaShop використовує JavaScript вже для клієнтської частини та покращення динамічної взаємодії на стороні клієнта та користувачького інтерфейсу [18]. За допомогою JavaScript можна використати AJAX, який дозволяє здійснювати асинхронні запити до

сервера без необхідності перезавантаження сторінки, що покращує продуктивність та швидкість роботи плагіна. Хоч і основна логіка та управління даним плагіном залишається в PHP, забезпечуючи сумісність з архітектурою PrestaShop.

Важливим аспектом є використання HTML та CSS стилів для наповнення плагіна та додавання стилів. Мова розмітки HTML використовується для створення основної структури веб-сторінок, включаючи формати запитань та відповідей, навігаційні елементи та інші компоненти інтерфейсу [19]. В свою чергу CSS використовуються для стилізації плагіна, які дозволяють створити привабливий та доступний дизайн інтерфейс плагіна.

Отже, вибір PHP як основи у розробці плагіна “Часті питання” для CMS PrestaShop обумовлений його сумісністю та інтеграцією з платформою, високою продуктивністю, можливостями безпеки та широкою підтримкою спільноти. Далі вже використання JavaScript забезпечує плагіну динамічну структуру і в свою чергу HTML та CSS надають наповнення та вузлізацію плагіна. У комплексі дані мови програмування та каскадна таблиця стилів CSS дозволяють розробити ефективний, зручний у використанні та доступний в обслуговуванні плагін для CMS PrestaShop.

2.3 Вибір середовища розробки

Вибір середовища розробки є важливим аспектом при розробці плагіна “Часті питання”. Середовище розробки відіграє ключову роль у процесі розробки якісного та надійного плагіна для такої платформи як платформа PrestaShop. Тому, для розробки плагіна було використано інтегроване середовище розробки PHPStorm.

Середовище розробки PHPStorm – це потужний інструмент, який спеціально розроблений для мови програмування PHP. Дане середовище розробки пропонує широкий набір інструментів, розширень та функцій, що забезпечують високій продуктивності та можливостям для розробки та високій

якості коду [20]. Серед переваг даного IDE середовища є інтелектуальний редактор для коду, який забезпечує автоматичне завершення коду, синтаксичне підсвічування, перевірку помилок у режимі реального часу та багато корисних функцій. Така можливість значно полегшує процес написання коду, зменшує кількість помилок та підвищує загальну ефективність розробки.

PHPStorm також забезпечує зручну роботу з базами даних, що є важливим аспектом для розробки плагіна для PrestaShop платформи. Інструменти для управління базами даних дозволяють легко створювати, редагувати SQL-запити, що забезпечує візуалізацію структури бази даних [21]. Це допомагає надати правильну інтеграцію плагіна з базою даних та зберегти її цілісність.

Отже, вибір такого середовища розробки як PHPStorm обґрунтований важливим критеріями. Це середовище задовільняє усі необхідні вимоги при ефективній розробці плагіна, що дозволяє надати високу якість та надійність продукту. PHPStorm підвищує продуктивність розробки, зменшує кількість помилок у коді та забезпечує простоту використання, що важливо для успішної реалізації проекту.

2.4 Взаємодія з ядром “PrestaShop”

Для проєктування плагіна платформа PrestaShop вимагає розуміння архітектури цієї платформи за для сумісності плагіна та розширення її механізмів. Взаємодія з ядром PrestaShop є суттєвим елементом при розробці, яка надає можливість інтеграції даного функціоналу з вже існуючими розширеннями та можливостями CMS платформи PrestaShop.

Основна реалізація інтегрування плагінів у PrestaShop залежить від системи хуків, які надають модулю можливість розробляти свій код у зазначених точках роботи самого ядра платформи. Для плагіна “Часті питання” критично важливо використовувати відповідні хуки для розробки, що і дасть змогу відобразити вкладку частих питань на відповідних сторінках інтернет-магазину, таких як сторінки продуктів, категорій та домашньої сторінки [22].

Також важливим аспектом за для сумісності з ядром PrestaShop є управління базою даних. Задля маніпуляцій над усіма даними, зберіганням питань та відповідей на них необхідно створити таблиці в базі даних, в яких і буде знаходитися вся інформація стосовно плагіна. Необхідним є використання методів інсталяції та деінсталяції плагіна, які й відповідно надають можливість створення таблиць при встановленні плагіна та їх видалення при видаленні плагіна. Це забезпечує чистоту бази даних та мінімізує ризики конфліктів з іншими модулями.

При взаємодії з ядром PrestaShop плагіну забезпечується додавання нових сторінок для платформи та керуванням плагіна, використовуючи стандартні можливості PrestaShop для розробки інтерфейсів та обробки запитів, які надходять. Це в свою чергу включає налаштування контролерів, форм та відображення контенту, що повинно відповідати основним та загальним стандартам даної платформи.

Тому, взаємодія з ядром PrestaShop при розробці плагіна “Часті питання” включає використання хуків, взаємодію з базою даних, розробку адміністративної частини та забезпечення сумісності з іншими модулями. Це забезпечує інтеграцію плагіна з платформою та надає користувачам та адміністраторам зручний та функціональний інструмент для роботи з поширеними питаннями.

2.5 Визначення акторів та діаграми варіантів використання

Після визначення основних вимог щодо розробки та проектування плагіна спершу необхідно визначити акторів та їх можливі варіанти використання. Серед акторів будуть: користувач(клієнт) та адміністратор.

Що стосується актора “Користувач” варіантами використання є “Перегляд продуктів”, “Перегляд категорій продуктів”, “Відображення сторінки продукту” та “Перегляд вкладки “Часті питання” на сторінці продукту/категорій” та “Можливість задати власне питання у вкладці “Часті питання””.

Актор “Користувач” отримує доступ до варіантів використання за допомогою Front-Office магазину, на якому може переходити по категоріях продуктів, переглянути там “Часті питання”, перейти до конкретного продукту та знайти часті питання користувачів до цього продукту. Також у актора “Користувач” є можливість задати своє власне питання, якщо зі списку запропонованих частих питань він не знайшов своєї відповіді.

На основі даного аналізу було створено діаграму використання для актора “Користувач”, яку зображено на рисунку 2.1.

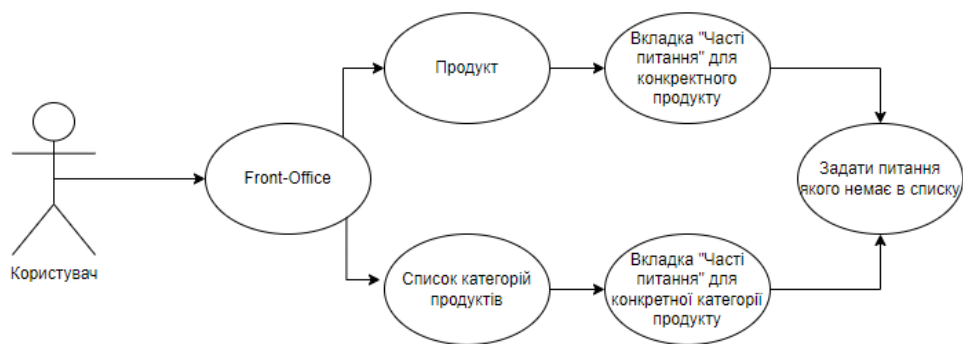


Рисунок 2.1 – Діаграма варіантів використання актора “Користувач”

Наступним етапом було описано варіанти використання та створення діаграми варіантів використання для актора “Адміністратор”.

Варіантами використання актора “Адміністратор” є “Перегляд адміністративної панелі плагіна”, “Доступ до загальних налаштувань”, “Відображення сторінки продукту з боку адміністратора ” та “Налаштування вкладки “Часті питання” на сторінці продукту/категорій з боку адміністратора ” та “Налаштування форми для питань користувачів””. На рисунку 2.2 зображено створену діаграму варіантів використання для актора “Адміністратор”.

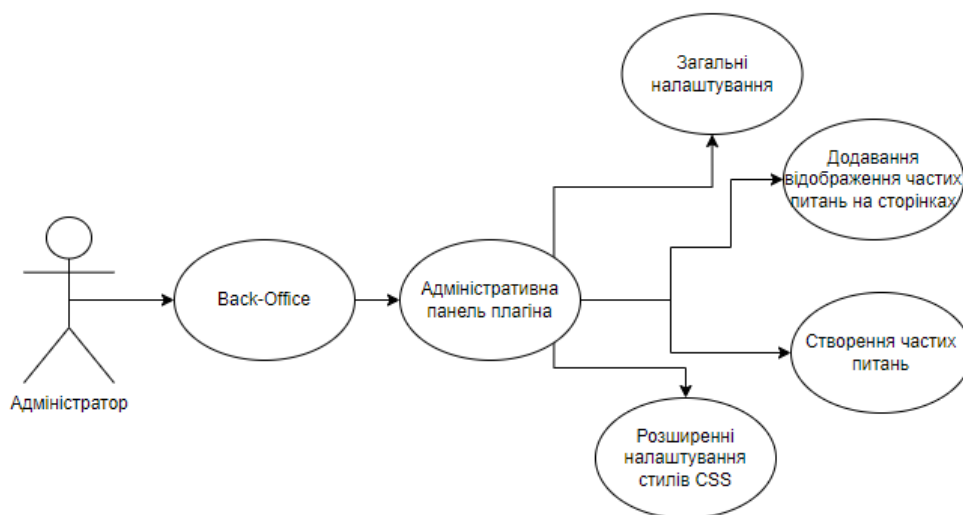


Рисунок 2.2 – Діаграма варіантів використання актора “Адміністратор”

Актор “Адміністратор” отримує доступ до варіантів використання за допомогою Back-Office магазину, на якому може налаштовувати часті питання по категоріях продуктів та по сторінках продуктів та можливість створення частих питань для конкретних продуктів. Також у актора “Адміністратор” є можливість створити форму для власного питання користувача, на яке відповідь він не знайшов. До того ж актор “Адміністратор” може змінювати стилі CSS для відображення вкладки “Часті питання” та редагувати показ питань на сторінках для актора “Користувач”.

2.6 Формування архітектури плагіна “Часті питання”

Проаналізувавши вимоги до плагіна, вибору мови програмування, аналізу предметної області та описів акторів та варіантів використання було спроектовано формування архітектури плагіна “Часті питання”. Під час формування було виділено трьохрівневу архітектуру, яка містить рівень клієнта, структурно-логічний рівень та рівень даних (див. рисунок 2.3).

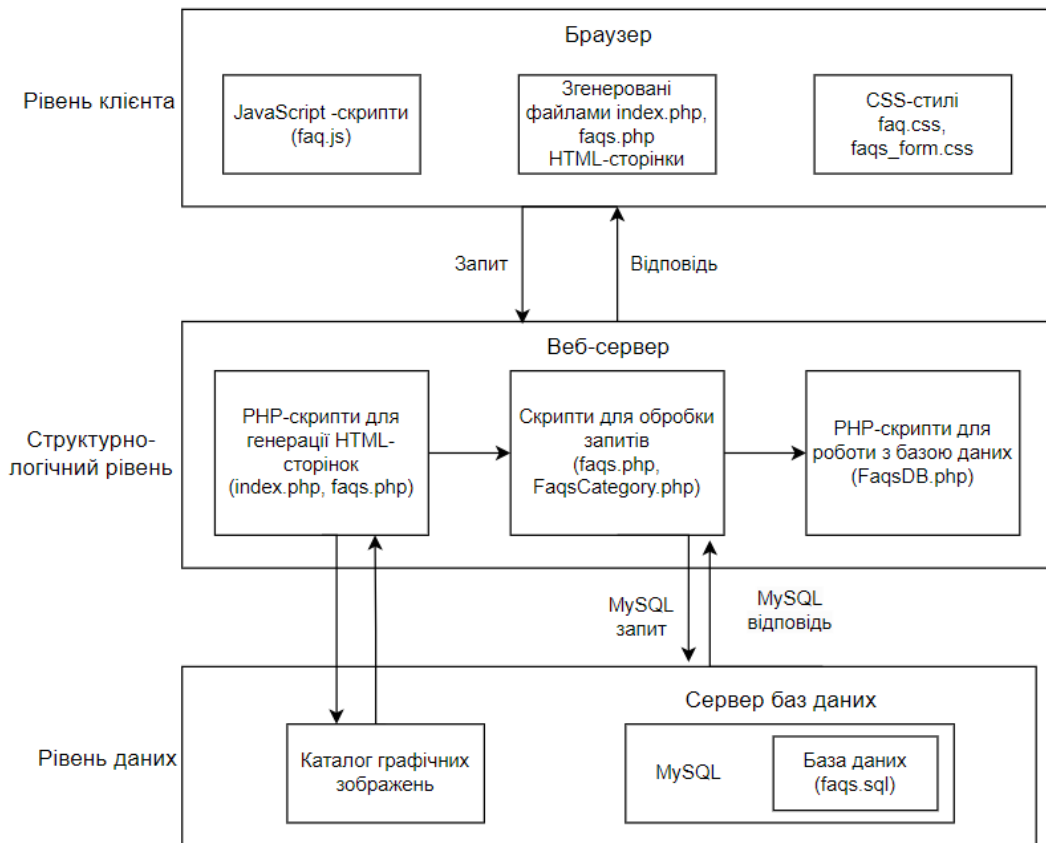


Рисунок 2.3 – Архітектура плагіна “Часті питання”

Зобразивши архітектуру плагіна “Часті питання” було проаналізовано кожен з рівнів. Що стосується рівня клієнта, то в ньому представлено все, що відображається у браузері, все це відображається за допомогою графічного інтерфейсу, який відображає вже згенеровані PHP-скриптами HTML-сторінки з елементами JavaScript [24].

На структурно-логічному рівні розміщений веб-сервер плагіна, який забезпечує весь функціонал програмних компонентів. На даному рівні представлена здебільшого структура самого сервісу та функціональних засобів, які відповідають за вивід HTML-сторінок, програмні скрипти PHP, їх алгоритми для взаємодії користувача з плагіном та уся обробка даних, яка надходить від користувачів.

Третім рівнем є рівень даних, який представляє сервер бази даних, який надає можливість опрацювання та зберігання даних, які містить плагін та інформації, яка надається користувачами. Основна функція рівня даних є

взаємодія з базою даних СКБД MySQL, від якої залежать всі операції над отриманими даними та їх вмістом [25].

2.7 Проєктування бази даних

Провівши аналіз щодо архітектури плагіна, визначення основних функціональних вимог при розробці, визначення акторів та діаграм використання наступним надважливим етапом є створення моделі БД для плагіна “Часті питання”[26].

Отже, необхідно виділити такі сутності:

- faq – визначає відомості про окремі питання.
- faq_category – містить інформацію про категорії частих питань.
- faq_shop – відповідає за інформацію про магазин та питань, які відносять до нього.
- faq_lang – визначає інформацію про часті питання на різних мовах.
- faq_category_shop – відповідає за зв’язок між категорією частих питань та магазином, до якого вони належать.
- faq_category_lang – відповідає за зв’язок між категорією частих питань та мовою, до якої вони належать.
- faq_association_product – містить зв’язок між конкретним продуктом та створеним питанням до нього.
- faq_association_category – містить зв’язок між певною категорією та створеним питанням до цієї категорії.

Спроектовану модель БД плагіна “Часті питання”, зображено на рисунку 2.4.

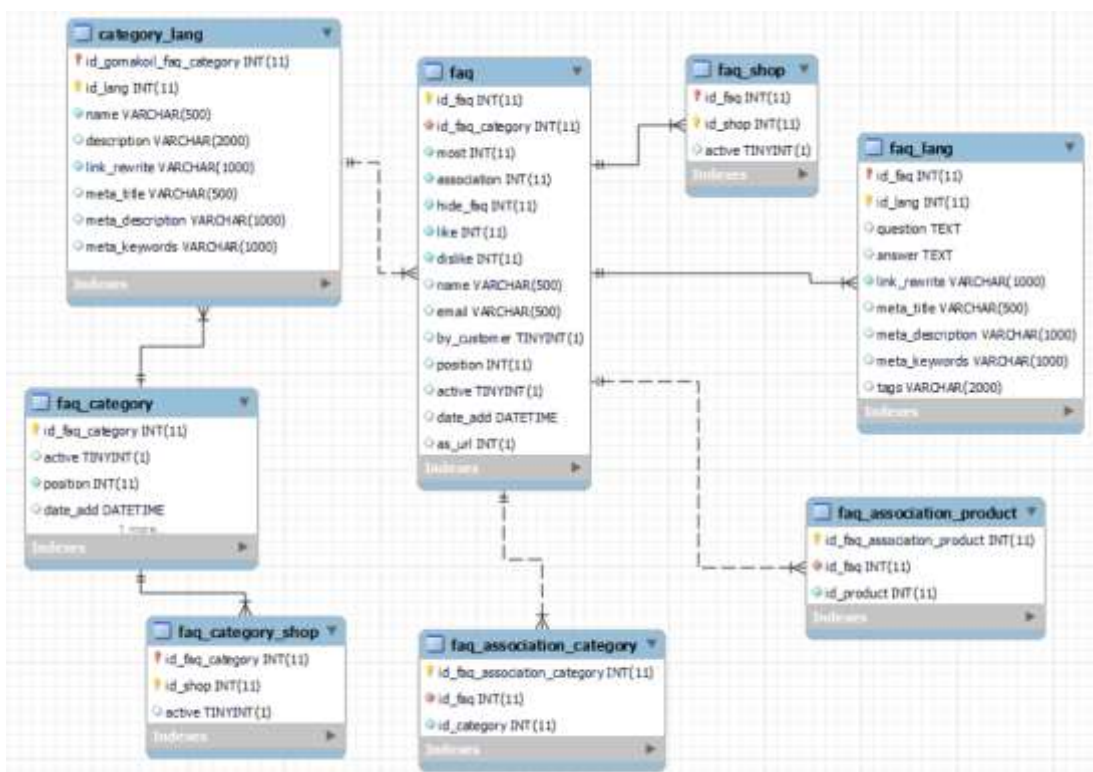


Рисунок 2.4 – Модель сутностей бази даних

Виділені сутності мають свої властивості, набори атрибутів та зв'язки відносно інших таблиць. У таблиці 2.1 представлено опис атрибутів відносно сутності “faq”.

Таблиця 2.1 – Опис атрибутів сутності “faq”

Атрибут	Тип даних	Призначення
1	2	3
id_faq	int(11)	ID питання (первинний ключ)
id_faq_category	int(11)	ID категорії (зовнішній ключ)
most	int	Відстеження найбільш популярних питань
association	int	Асоціація з іншими елементами

Продовження таблиці 2.1

1	2	3
hide_faq	int	Приховування товару
like	int	Кількість лайків
dislike	int	Кількість дизлайків
name	nvarchar(500)	Ім'я автора питання
email	nvarchar (500)	Електронна пошта автора питання
by_customer	boolean	Чи питання створене клієнтом
position	INT	Позиція питання
active	boolean	Активне питання
date_add	datetime	Дата та час створення питання
as_url	int(1)	Поле для відстеження URL асоціації

В таблиці 2.2 зображено перелік та опис атрибутів відносно сутності “faq_category”.

Таблиця 2.2 – Опис атрибутів сутності “faq_category”

Атрибут	Тип даних	Призначення
id_faq_category	int(11)	ID категорії (PK)
active	boolean	Активне категорія
position	int	Позиція категорії
date_add	datetime	Дата та час створення категорії
color	text	Колір, який асоціюється з категорією

В таблиці 2.3 зображено перелік та опис атрибутів відносно сутності “faq_shop”.

Таблиця 2.3 – Опис атрибутів сутності “faq_shop”

Атрибут	Тип даних	Призначення
id_shop	int(11)	ID магазину (первинний ключ)
id_faq	int(11)	ID питання (зовнішній ключ)
active	boolean	Активне питання в конкретному магазині

В таблиці 2.4 подано перелік та опис атрибутів відносно сутності “faq_lang”.

Таблиця 2.4 – Опис атрибутів сутності “faq_lang”

Атрибут	Тип даних	Призначення
1	2	3
id_lang	int(11)	ID мови (первинний ключ)
id_faq	int(11)	ID питання (зовнішній ключ)
question	text	Текст питання на конкретній мові
answer	text	Текст відповіді на конкретній мові.
link_rewrite	nvarchar(1000)	Переписане посилання для SEO на конкретній мові.

Продовження таблиці 2.4

1	2	3
meta_title	nvarchar (500)	Мета-заголовок для SEO на конкретній мові.
meta_description	nvarchar (1000)	Мета-опис для SEO на конкретній мові.
meta_keywords	nvarchar (1000)	Мета-ключові слова для SEO на конкретній мові.
tags	nvarchar (2000)	Теги для питання на певній мові

У таблиці 2.5 представлено опис атрибутів сутності “faq_category_shop”.

Таблиця 2.5 – Опис атрибутів сутності “faq_category_shop”

Атрибут	Тип даних	Призначення
id_shop	int(11)	ID магазину
id_faq_category	int(11)	ID категорії (зовнішній ключ)
active	boolean	Активне питання

У таблиці 2.6 представлено опис атрибутів сутності “faq_category_lang”.

Таблиця 2.6 – Опис атрибутів сутності “faq_category_lang”

Атрибут	Тип даних	Призначення
1	2	3
id_lang	int(11)	ID мови
id_faq_category	int(11)	ID категорії (зовнішній ключ)

Продовження таблиці 2.6

1	2	3
name	nvarchar(500)	Назва категорії на конкретній мові
description	nvarchar(2000)	Опис категорії на конкретній мові
link_rewrite	nvarchar(1000)	Переписане посилання для SEO на конкретній мові.
meta_title	nvarchar(500)	Мета-заголовок для SEO на конкретній мові.
meta_description	nvarchar(1000)	Мета-опис для SEO на конкретній мові.
meta_keywords	nvarchar(1000)	Мета-ключові слова для SEO на конкретній мові.

У таблиці 2.7 представлено опис атрибутів сутності “faq_association_product”, серед яких атрибутивати: id_association_product, id_faq та id_product.

Таблиця 2.7 – Опис атрибутів сутності “faq_association_product”

Атрибут	Тип даних	Призначення
id_association_product	int(11)	ID асоціації питання з продуктом (первинний ключ)
id_faq	int(11)	ID питання (зовнішній ключ)
id_product	int(11)	ID продукту

У таблиці 2.8 представлено опис атрибутів сутності “faq_association_category”.

Таблиця 2.8 – Опис атрибутів сутності “faq_association_category”

Атрибут	Тип даних	Призначення
id_association_category	int(11)	ID асоціації питання з категорією (первинний ключ)
id_faq	int(11)	ID питання (зовнішній ключ)
id_category	int(11)	ID категорії

Відобразивши усі атрибути сутностей було визначено зв'язки між даними сутностями та таблицями [27]. Таблиця “faq” має зв'язок один до багатьох із таблицями “faq_lang”, “faq_shop”, “faq_association_product”, “faq_association_category”. Це обумовлено тим, що кожне питання може бути мати декілька мов та може бути пов'язане з кількома категоріями або ж продуктами [28]. Що стосується таблиці “faq_category”, то вона має зв'язок один до багатьох із таблицями “faq_category_shop”, “faq_category_lang” та “faq”. Так як кожна категорія може містити кілька частих питань і може бути прив'язана до кількох магазинів та мати кілька мов відповідно.

2.8 Проєктування структури функціональних компонентів

Необхідним та важливим етапом при розробці плагіна “Часті питання” є визначення та проєктування функціональних компонентів, зв'язків та залежностей між цими зв'язками. На рисунку 2.5 зображено структуру компонентів на визначення принципів їх взаємодії між собою для плагіна “Часті питання”.

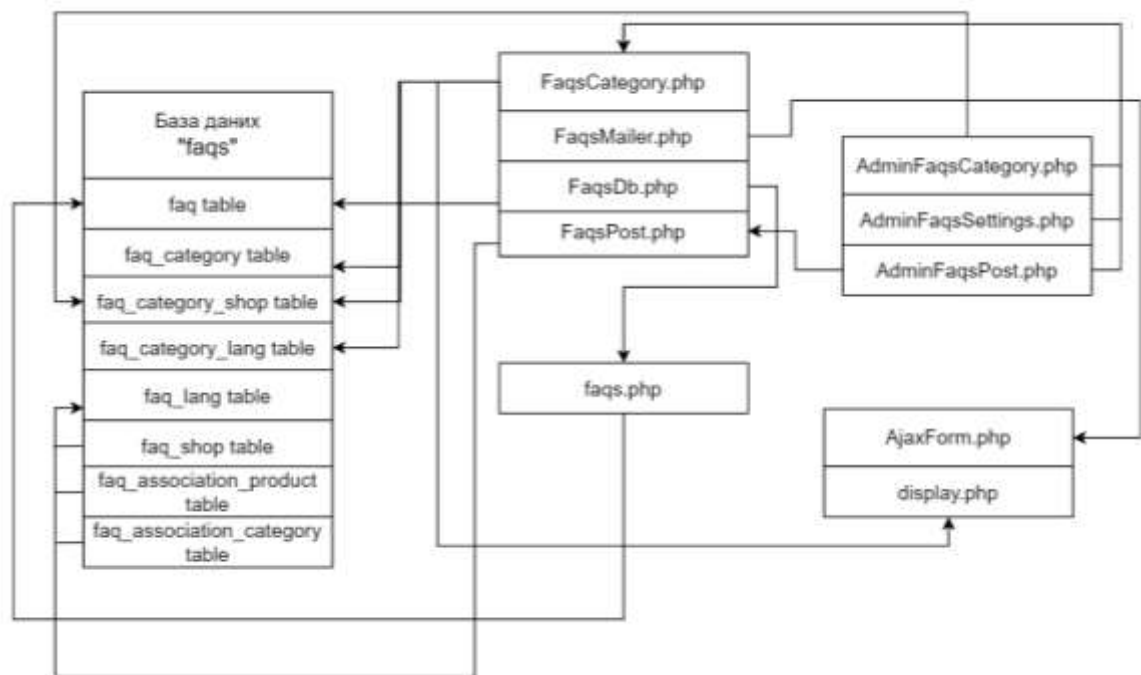


Рисунок 2.5 – Схема проєктування функціональних компонентів

Спершу було розглянуто такий компонент як “faqs.php”. Файл “faqs.php” є одним із основних файлів плагіна, який визначає назву та версію модуля, встановлює шлях до шаблонів. Також в даному файлі відбувається реєстрація хуків, до яких підключається модуль для виконання всіх функцій, встановлюються залежності відносно версії PrestaShop та їх сумісності.

В основі структури веб-сторінок для реалізації плагіна знаходяться такі основні файли як: “FaqsCategory.php” та “AdminFaqsCategory.php”, які відповідають за наповнення сторінок категорій як з боку користувача так і з боку адміністратора. Користувач може перейти у конкретну категорію продуктів та знайти питань та відповідь до них, в свою чергу адміністратор зможе редагувати інформацію, фільтрувати та видаляти ту, яка стосується категорій і в даному випадку для дій над контентом категорій уся інформація записується у таблиці баз даних faq_category, faq_association_category, faq_category_shop та faq_category_lang. “FaqsPost.php” – файл, який відповідає за функціонал сторінки частого питання, його вивід, яку може побачити користувач.

Якщо користувач не знайшов відповіді на своє питання, він може задати власне питання, скориставшись формою. При натисканні кнопки “Задати

питання” відбувається відкриття форми за допомогою js-компонентів та файлу AjaxForm.php, який відповідає за наповнення форми та надсилання даних.

Доступ адміністратора до компоненту “AdminFaqsPost.php” може отримати за допомогою входу на сторінку конфігурації плагіна, та перейшовши до вкладки, що стосується створення частого питання, за що й відповідає компонент “AdminFaqsPost.php”.

За допомогою проектування структури функціональних компонентів було описано та проаналізовано які дії виконують компоненти, та як вони взаємопов’язані з іншими файлами, які забезпечують цілісність реалізації та роботи плагіна “Часті питання”.

2.9 Файлова структура плагіна “Часті питання”

Проектування узагальненої файлової структури плагіна є вагомим етапом у розробці так як забезпечує логічну та структуровану організацію коду [29]. Для плагіна “Часті питання” файлову структуру необхідно проектувати відповідно до основних вимог та стандартів PrestaShop, що й забезпечуватиме сумісність з CMS PrestaShop. На рисунку 2.6 зображено файлову структуру плагіна “Часті питання”.

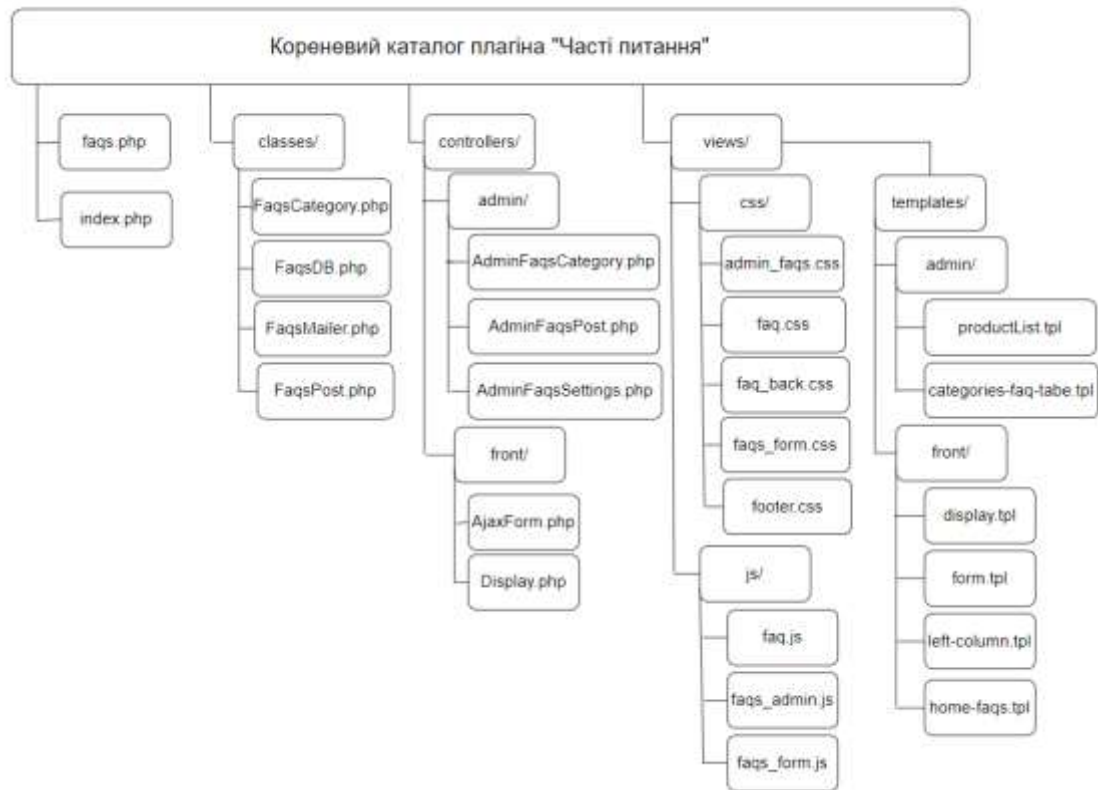


Рисунок 2.6 – Файлова структура плагіна “Часті питання”

Було проведено детальний огляд файлової структури та розглянуто призначення файлів. Файли “index.php” та “faqs.php” знаходяться в кореневому каталозі.

– “index.php” – виконує функцію захисту каталогу від несанкціонованого доступу. До того ж такий файл є стандартним для плагінів у PrestaShop та їх безпеки.

– “faqs.php” – виконує функцію забезпечення функціональності плагіна. В даному файлі здійснюється реєстрація хуків, додавання HTML, CSS та JS файлів.

У плагіні також створено інші підкаталоги. Спершу розглянемо наповнення

Каталогу “classes/”, який містить файли:

– “FaqsCategory.php” – виконує усі маніпуляції стосовно категорій: з таблицею баз даних, вмістом, полями сторінки.

– “FaqsDB.php” – виконує створення таблиць баз даних, їх редагування та видалення.

–“FaqsMailer.php” – забезпечує розсилку повідомлень користувачам та адміністратором, коли користувач створює у формі власне питання, щоб отримати на нього відповідь.

–“FaqsPost.php” – виконує усі маніпуляції стосовно створеного поширеного питання та його сторінки.

Наступним каталогом є каталог “controllers”, який в свою чергу розбивається на 2 підкаталоги “admin” та “front” відповідно. Підкаталог “admin” містить в свої такі елементи:

–“AdminFaqsCategory.php” – відповідає за вивід та наповнення сторінки категорій поширених питань, та налаштувань категорій.

–“AdminFaqsPost.php” – відповідає за вивід та наповнення сторінки категорій поширених питань з боку адміністратора.

–“AdminFaqsSettings.php” – забезпечує вивід та наповнення сторінки налаштувань адміністратора відносно плагіна.

В свою чергу підкаталог “front” містить елементи такі як:

–“AjaxForm.php” – відповідає за створення форми для можливості задання власного питання користувачем. В даному файлі відбувається отримання даних переданих користувачем для адміністратора та відповідь на них.

–“Display.php” – слугує для відображення усього контенту, що відобразатиметься збоку користувача інтернет-магазину та плагіна.

У каталозі “views/” є підкаталоги “css/”, “js/” та “templates/”. Підкаталог “css/” містить в собі файли, які відповідають за дизайн сторінки, їх стилі, а саме:

–“admin_faqs.css” – відповідає за відображення піктограми в адміністративній панелі плагіна.

–“faq.css” – структура для надання стилів для повної сторінки плагіна “Часті питання” на платформі.

–“faq_back.css” – відповідає за стилізацію адміністративної частини плагіна “Часті питання”.

–“faqs_form.css” – стилі, для відображення форми для запитань у плагіні.

–“fotter.css” – відповідає за стилі в нижній частині сторінки.

Підкаталог “js/” містить в собі такі файли:

–“faq.js” – функціонал для динамічної взаємодії користувача із плагіном в інтернет-магазині.

–“faqs_admin.js” – функціонал для обробки дій, які виконуються в налаштуванні плагіна в адміністративній панелі.

–“faqs_form.js” – відповідає за функціонал форми для задання питання.

Також каталог “views/” містить підкаталог “templates/”, який розділений на “admin/” та “front/”.

Підкаталог “admin/” містить файли:

–“productList.tpl” – відповідає за відображення списку доданих продуктів у вигляді таблиці в адміністративній панелі модуля, що в свою чергу дозволяє додавати часті питання до продуктів.

–“categories-faqs-table.tpl” – шаблон, який відповідає за відображення списку категорій для плагіна.

Підкаталог “front/” містить файли:

–“display.tpl” – функціонал для відображення сторінки з частими питаннями.

–“form.tpl” – функціонал для створення інтерфейсу для форми, за допомогою якої користувач може задати питання, на яке не знайшов відповіді.

–“left-column.tpl” – відповідає за відображення вмісту блоку “Часті питання” в лівій колонці сторінок.

–“home-faqs.tpl” – шаблон, який відповідає за відображення вмісту блоку “Часті питання” на сторінці “Home”.

Отже, в даному підрозділі було визначено структуру каталогів для плагіна “Часті питання” та короткий опис вмісту кожного файлу, який відповідає за певний функціонал для цілісності проектування та в подальшому розробки плагіна на базі CMS PrestaShop.

2.10 Висновок до другого розділу

У другому розділі було виконано комплексний аналіз для проєктування плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop. Спершу було визначено основні вимоги для плагіна, які повинні бути враховані за для ефективної розробки та задоволення усіх потреб користувачів.

Важливим аспектом було врахування сумісності модуля з ядром PrestaShop, яке надасть інтеграцію плагіна на платформу PrestaShop без порушення та уникнення помилок до функціоналу роботи вже існуючих розширень на платформі. До того ж, було спроектовано файлову структуру плагіна та структуру функціональних компонентів, які забезпечать надійність функціоналу та відповідності усім вимогам сучасної електронної комерції для інтрнет-магазинів.

Тому, проєктування та в подальшому розробка даного плагіну “Часті питання” покращать функціонал роботи магазину та підвищать рівень обслуговування та задоволенності клієнтів під час використання, так як це дасть змогу швидко знаходити користувачам відповіді на їх питання, а адміністраторам полегшить роботу підтримки у відповіді на часті питання відносно продуктів.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПЛАГІНА “ЧАСТІ ПИТАННЯ”

3.1 Розробка плагіна “Часті питання”

Розробка плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop є головним етапом при створенні та реалізації плагіна. На даному етапі необхідно підготувати середовище розробки, реалізувати основні функції плагіна та провести його інтеграцію з користувацьким інтерфейсом.

3.1.1 Підготовка середовища розробки

Для успішної розробки плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop необхідно правильно підготувати середовище розробки. Це включає налаштування програмного забезпечення та інструментів, які забезпечать ефективний та безперебійний процес розробки.

Першим кроком у підготовці середовища розробки є встановлення PrestaShop. Для цього необхідно завантажити останню версію PrestaShop з офіційного сайту та розпакувати її у відповідний каталог на сервері. Потім слід створити нову базу даних у MySQL та пройти через процес встановлення PrestaShop, дотримуючись інструкцій щодо встановлення.

Після успішного встановлення PrestaShop необхідно налаштувати середовище розробки. Для цього потрібно встановити інтегроване середовище розробки (IDE), таке як PhpStorm, яке підтримує розробку на PHP та має інструменти для роботи з HTML, CSS та JavaScript. Цей IDE забезпечує зручний інтерфейс для написання коду, налагодження та тестування плагінів.

Таким чином, підготовка середовища розробки є важливим етапом, що включає налаштування веб-сервера, встановлення PrestaShop, вибір та налаштування IDE, використання системи контролю версій та підготовку інструментів для тестування. Всі ці компоненти забезпечують ефективний

процес розробки, тестування та впровадження плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop.

3.1.2 Реалізація основних функцій плагіна

Реалізація основних функцій плагіна “Часті питання” включає кілька ключових етапів: розробку адміністративної частини для управління питаннями та відповідями, створення користувацького інтерфейсу для відображення частих питань на сторінках магазину та забезпечення інтеграції плагіна з існуючими модулями PrestaShop.

Спершу було розроблено адміністративний інтерфейс плагіна. Це включає створення окремої сторінки в панелі адміністратора PrestaShop, де адміністратори магазину зможуть додавати, редагувати та видаляти питання та відповіді. За цей етап відповідають розроблені файли `AdminFaqsSettings.php`, `AdminFaqsPost.php` та `AdminFaqsCategory.php`, в яких відбувається реалізація налаштувань, додавання питань та відповідей як на окрему сторінку частих питань так і на сторінки категорій. Для цього створюється новий контролер, який буде відповідати за обробку запитів до цієї сторінки, а також відповідні моделі та представлення.

Наступним кроком є розробка бази даних для зберігання питань та відповідей. Було створено таблиці в базі даних PrestaShop, яка буде містити поля для зберігання тексту питання, відповіді, дати створення та оновлення, а також інші необхідні метадані. Це забезпечить зручне зберігання та швидкий доступ до даних плагіна. Реалізація створення таблиць та додавання полів реалізовано у файлі `FaqsDB.php`, який і відповідає за створення таблиць, додавання оновлення та видалення.

За для забезпечення інтерактивність та покращення користувацького досвіду, було додано функціональність для пошуку по частих питаннях, а також можливість фільтрації питань за категоріями. Для цього реалізовано додаткові

методи в контролері, а також створено відповідні AJAX-запити для динамічного оновлення контенту на сторінці без її перезавантаження.

Інтеграція плагіна з іншими модулями та функціональністю PrestaShop є важливим аспектом реалізації. Було додано можливість відображення найбільш популярних питань та можливість реакції лайком або ж дизлайком на відповідне питання та відповідь на нього. Для цього потрібно використовувати хуки PrestaShop, які дозволяють вставляти власний код у визначені місця шаблонів магазину.

Таким чином, реалізація основних функцій плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop включає розробку адміністративного інтерфейсу, бази даних для зберігання питань та відповідей, створення користувацького інтерфейсу, а також забезпечення інтеграції з існуючими модулями та шаблонами. Це дозволяє створити функціональний та зручний у використанні плагін, який покращує взаємодію користувачів з інтернет-магазином та забезпечує швидкий доступ до необхідної інформації.

3.2 Огляд функціоналу CMS платформи PrestaShop

PrestaShop є однією з найбільш популярних систем управління контентом (CMS) для електронної комерції, що пропонує багатий набір функцій для створення та управління інтернет-магазинами. Вона поєднує в собі потужність, гнучкість та зручність використання, що робить її ідеальним вибором як для малих, так і для великих підприємств.

Однією з ключових особливостей PrestaShop є її модульна архітектура. Кожен функціональний компонент, такий як каталог товарів, управління замовленнями або система оплати, реалізований у вигляді окремого модуля. Це дозволяє розширювати та адаптувати функціональність магазину відповідно до конкретних потреб. Модулі можуть бути легко додані, видалені або налаштовані через адміністративну панель, що забезпечує високу гнучкість у управлінні

магазином. На рисунку 3.1 зображено інтерфейс платформи PrestaShop та сторінка додаткових плагінів для використання.

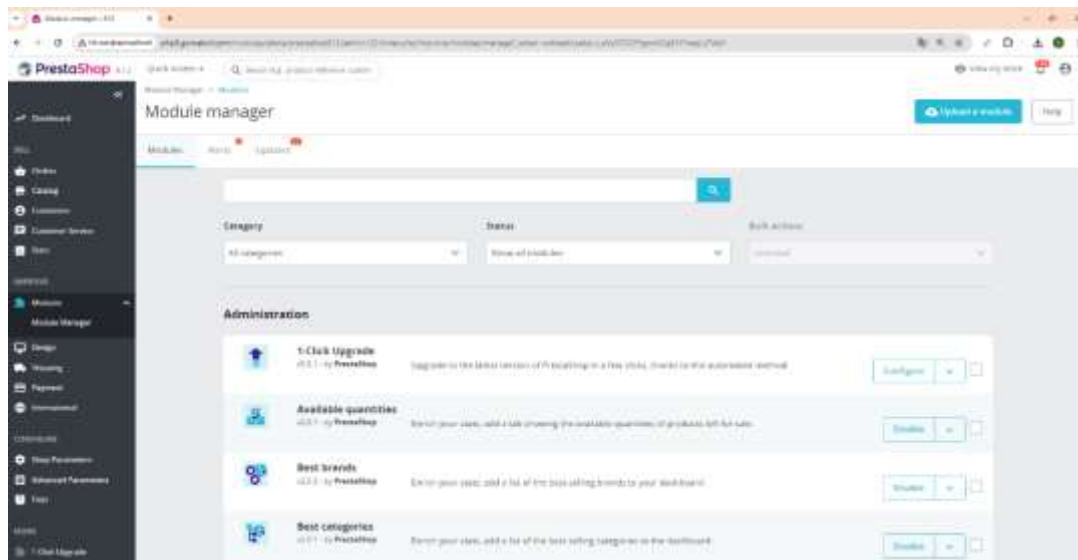


Рисунок 3.1 – Інтерфейс платформи PrestaShop

На платформі представлена бічна панель, яка містить вкладки замовлень, каталогу товарів, клієнтів, налаштувань плагінів, оплати, доставки та багато інших можливостей.

Слід зазначити, каталог товарів у PrestaShop, який здатен підтримувати необмежену кількість продуктів та категорій. Інтуїтивно зрозумілий інтерфейс для управління каталогом дозволяє швидко додавати нові товари, налаштовувати їхні атрибути, ціни, знижки та варіанти доставки (див. рисунок 3.2).

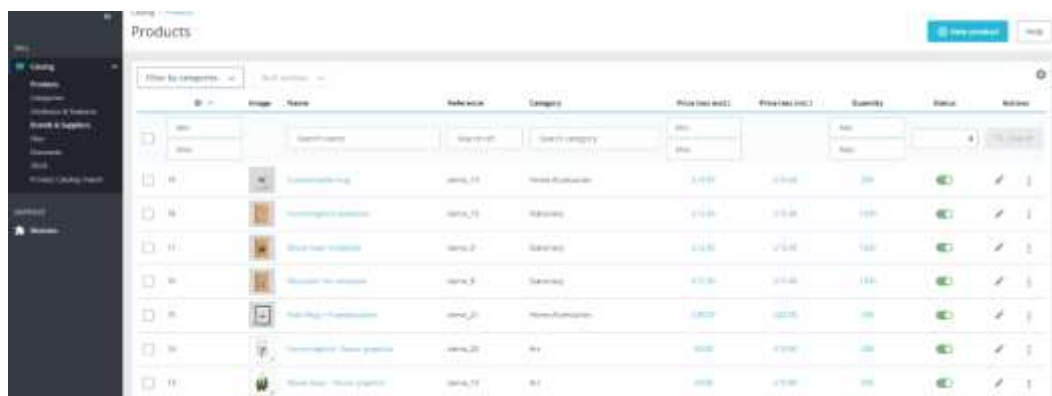


Рисунок 3.2 – Огляд каталогу товарів на платформі PrestaShop

Крім того, система підтримує функції SEO-оптимізації, що допомагає підвищити видимість товарів у пошукових системах.

CMS PrestaShop забезпечує широкий набір інструментів для управління замовленнями та клієнтами. Адміністратори можуть легко відстежувати статус замовлень, обробляти повернення та обміни, а також генерувати звіти про продажі та клієнтську активність. На рисунку 3.3. зображено сторінку замовлення на платформі.

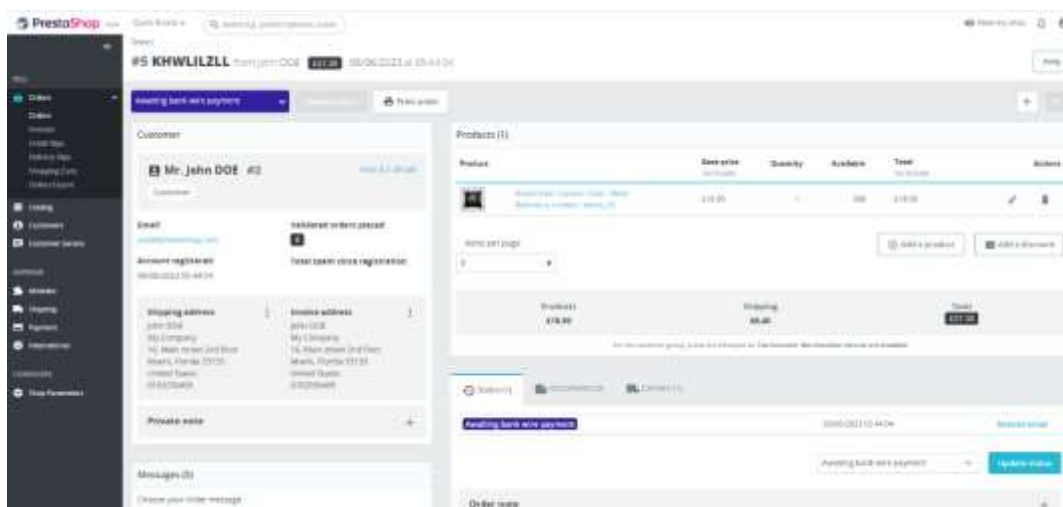


Рисунок 3.3 – Сторінка замовлення на платформі PrestaShop

В даній вкладці “Замовлення” адміністратор може здійснювати будь-які маніпуляції над замовленням. Починаючи від створення нового замовлення, закінчуючи редагуванням типу оплати, доставки та зміни статусу замовлення.

Однією з важливих складових PrestaShop є її аналітичні інструменти. Вони дозволяють відстежувати ключові показники продуктивності магазину, такі як обсяги продажів, середній чек, конверсії, відвідуваність та багато іншого. Ці дані можуть бути використані для прийняття обґрунтованих бізнес-рішень та оптимізації маркетингових стратегій (див. рисунок 3.4).

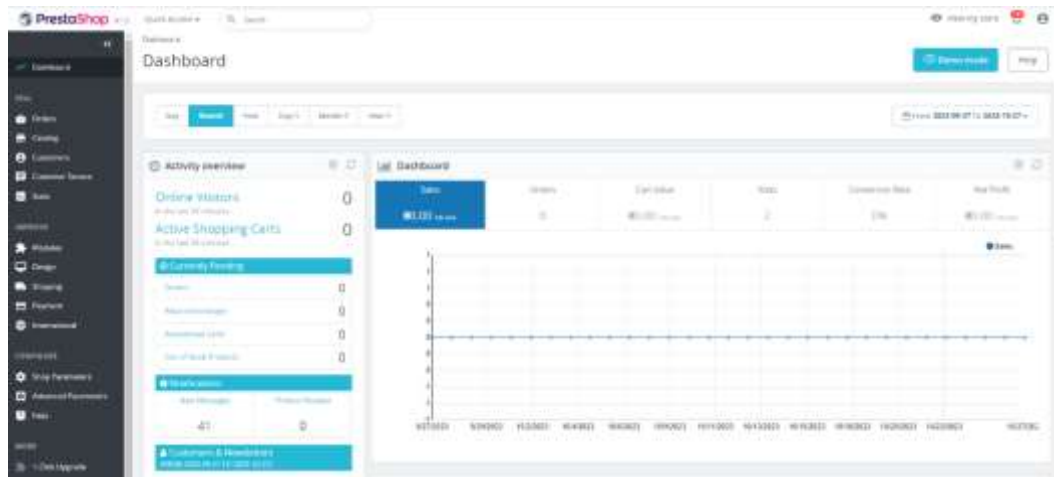


Рисунок 3.4 – Огля сторінки “Dashboard” на платформі PrestaShop

Крім основних функцій, PrestaShop підтримує широкий спектр додаткових можливостей через маркетплейс модулів та тем. Це дозволяє знайти та інтегрувати додаткові функціональні модулі, такі як інструменти для автоматизації маркетингу, інтеграції з соціальними мережами, системи лояльності та багато іншого.

Таким чином, PrestaShop надає повний набір інструментів для створення та управління інтернет-магазином, що дозволяє бізнесам ефективно вести на даній платформі електронну комерцію. Завдяки своїй модульній архітектурі, гнучкості та багатому функціоналу, PrestaShop є потужною платформою, яка може задовольнити потреби як новачків, так і досвідчених підприємців у сфері електронної комерції.

3.3 Впровадження плагіна в інтернет-магазинах

Впровадження плагіна “Часті питання” в інтернет-магазинах на базі CMS PrestaShop є завершальним етапом розробки, який охоплює інтеграцію та налаштування плагіна у реальних умовах. Цей процес дає змогу забезпечити безперебійну роботу всього функціоналу плагіна та його взаємодію з іншими елементами магазину.

3.3.1 Інструкція зі встановлення плагіна

Перш за все, перед впровадженням плагіна необхідно переконатися, що він сумісний з версією PrestaShop, яка використовується в інтернет-магазині. Це включає перевірку версійної сумісності PHP, MySQL та інших залежностей, що враховано при розробці плагіна “Часті питання”.

Наступним кроком є встановлення плагіна через панель адміністратора PrestaShop. На рисунку 3.5 зображено сторінку “Module Manager” через яку й відбувається завантаження плагіна на платформу PrestaShop.

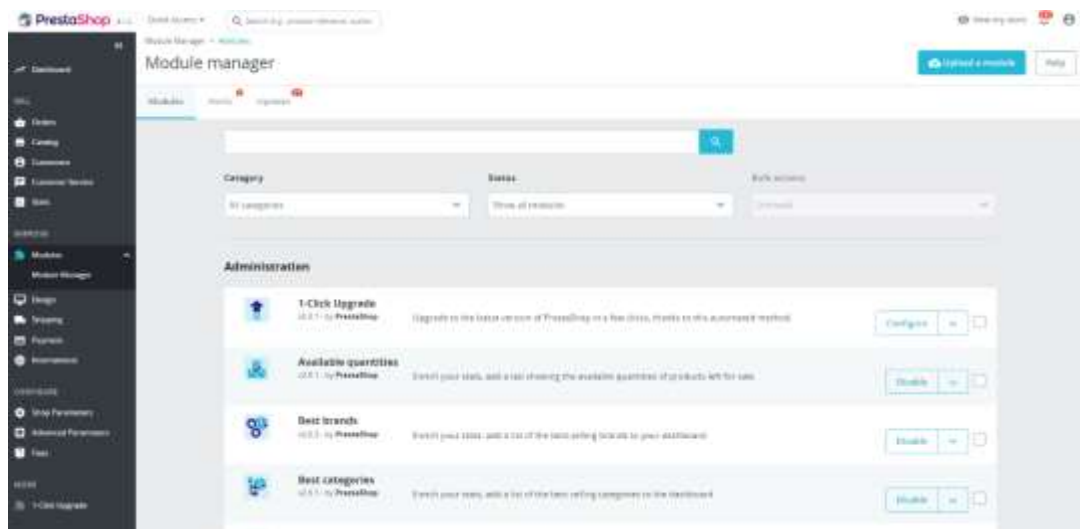


Рисунок 3.5 – Сторінка “Module Manager” на платформі PrestaShop

Щоб завантажити плагін на магазин необхідно увійти в адміністративну панель PrestaShop. Перейти на сторінку “Module Manager” та архів файлу розробленого плагіна завантажити через кнопку “Upload a module”. У вікні, що відкриється, необхідно обрати завантажений архів з плагіном та завантажити плагін на платформу. Це є стандартною опцією для встановлення плагінів, що включає завантаження архіву з плагіном та його активацію (див. рисунок 3.6).



Рисунок 3.6 – Встановлений плагін “Часті питання”

Якщо процес інсталяції виконано успішно, то можна буде побачити повідомлення про успішну інсталяцію плагіна. До того ж, якщо встановлення плагіна пройшло успішно, то він має з'явитися у списку серед вже встановлених та доступних модулів, де можна провести його базові налаштування.

3.3.2 Налаштування плагіна та огляд основних можливостей

Після встановлення необхідно налаштувати плагін відповідно до потреб конкретного інтернет-магазину. Це включає створення та додавання питань/відповідей, фільтрування частих питань по категоріях, їх налаштування відображення на фронтенді, створення форми для задавання власного питання та її оформлення. Таким чином гнучкість налаштувань дозволяє адміністраторам магазину адаптувати плагін під свої вимоги та забезпечити максимально комфортне використання для користувачів.

На рисунку 3.7 зображено інтерфейс плагіна сторінки “Settings”.

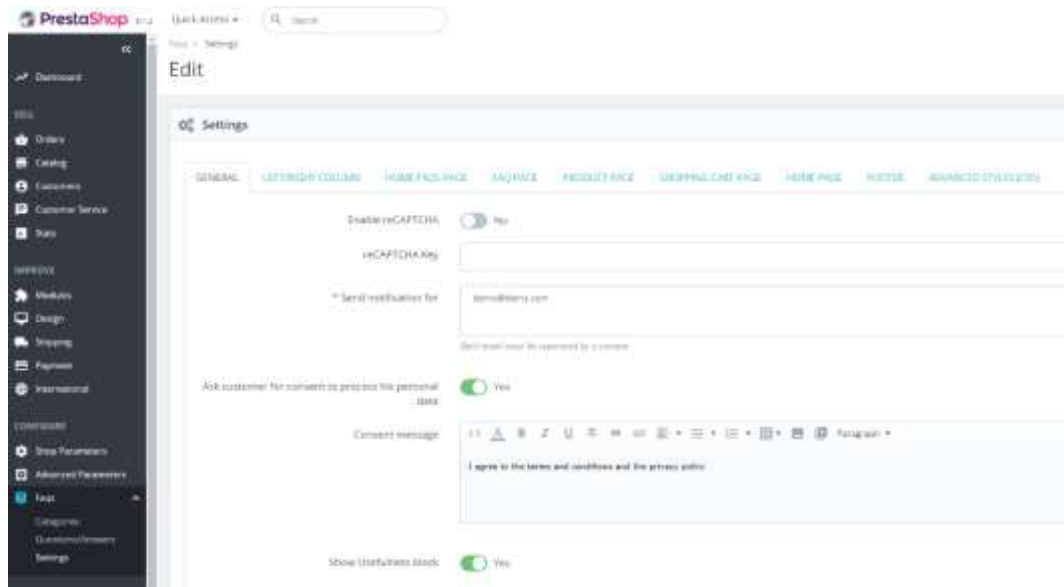


Рисунок 3.7 – Інтерфейс сторінки “Settings” плагіна

На даній сторінці зображено загальні налаштування для вкладки частих питань, конфігурація форми для задання власного питання клієнта. До прикладу поле “Send notification for” відповідає за надсилання на пошту сповіщення, якщо клієнт на магазині задав власне питання. Це допоможе швидко реагувати на потреби клієнтів та відповідати на їх запитання.

Наступною є вкладка “Left/Right Column”, яка відповідає за керування відображенням частих питань на сторінках інтернет-магазину збоку в колонках. Адміністратор може обирати чи відображати часті питання на сторінці категорій, на сторінці продукту в колонці чи взагалі відображати часті питання лише на сторінці “Частих питань”. Також є вибір чи буде доступною кнопка для задання власного питання чи вимкнуті її (див. рисунок 3.8)

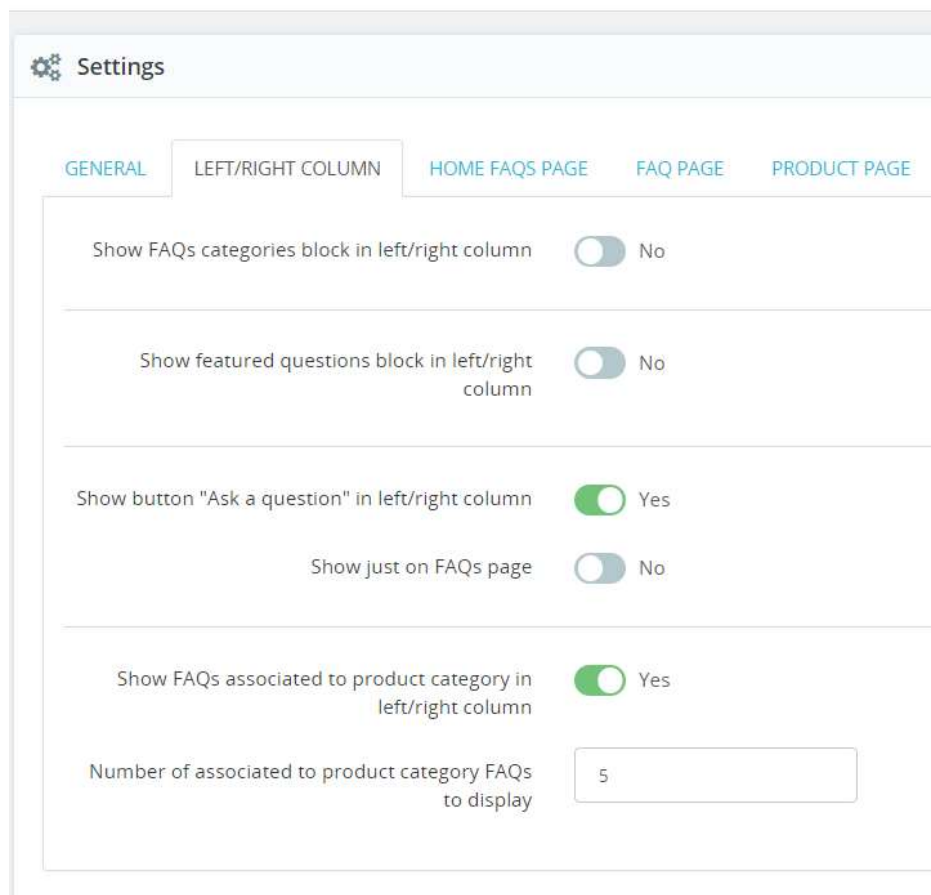


Рисунок 3.8 – Вкладка “Left/Right Column”

Для вкладок “Home Faqs Page”, “Faq Page”, “Product Page” використовуються подібні налаштування, але вже для конкретних сторінок. Тобто на сторінці “Faq Page” можна налаштувати відображення запитань, які будуть відображатися відповідно до категорій, чи надати можливість відображення питань та відповідей пов’язаних продуктів.

Виконавши основні налаштування стосовно вкладки частих питань, можна перейти до вкладки створення питань та відповідей до них. Так як плагін “Часті питання” на платформі знаходиться в лівому боковому меню, то щоб реалізувати етап створення частого питання, то необхідно перейти на сторінку “Questions/Answers” (див. рисунок 3.9).

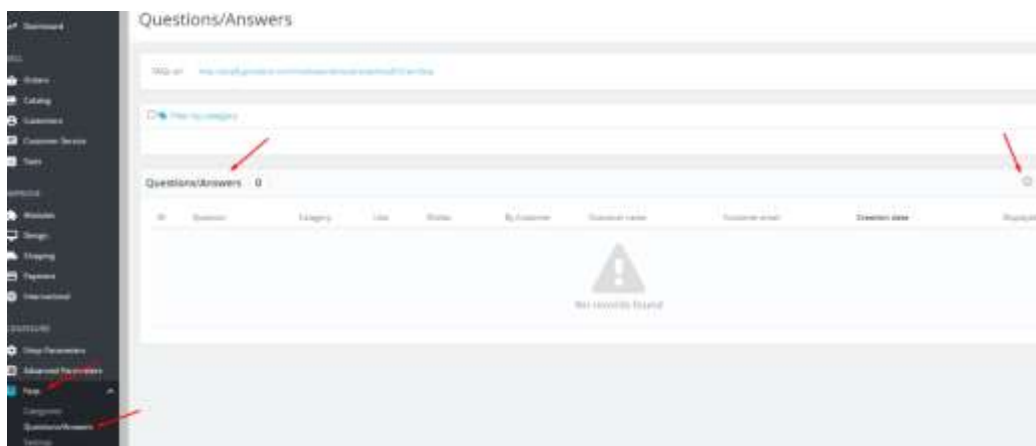


Рисунок 3.9 – Сторінка “Questions/Answers”

Дана сторінка надає можливість створювати конкретні часті питання та відповіді на них. Ті питання, які будуть створені, будуть відображатися в списку на цій же сторінці.

Нажавши на позначку плюс, що означає додати нове питання, відкривається сторінка створення частого питання та відповіді на нього (див. рисунок 3.10).

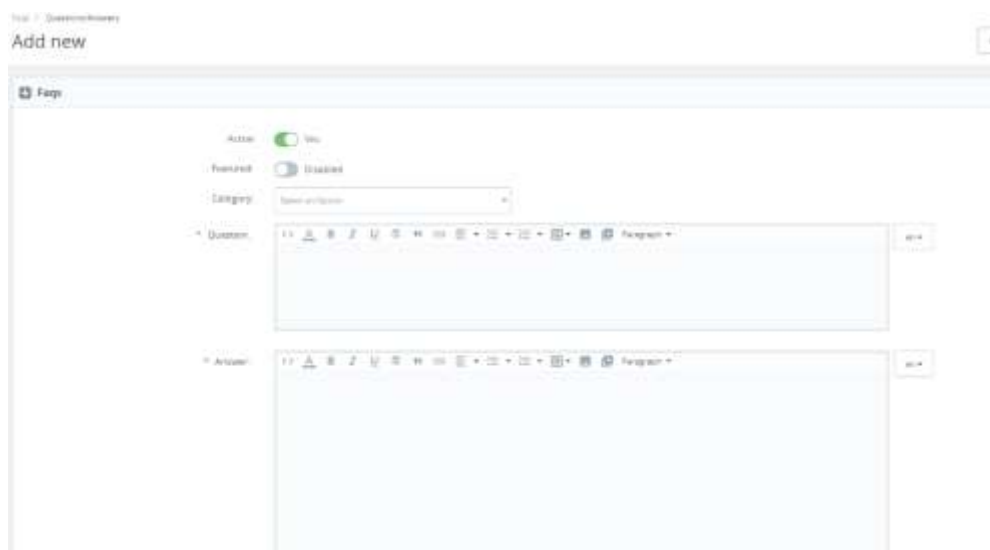


Рисунок 3.10 – Сторінка “Створення нового питання”

Дана сторінка дозволяє створити питання, надати одразу на нього відповідь. Також можна визначити чи буде дане питання активним, або ж якщо питання стало неактуальним, то його можна вимкнути. До того ж, є можливість

прив'язати це питання до конкретної категорії, де це питання й буде відображатися. За відображення даної сторінки в адміністраторській панелі відповідає файл AdminFaqsPost.php. Саме у функції renderForm(), створюється форма з полями “Questions”, “Answers” та формується асоціація з продуктами та та обрані категорії (див. додаток А).

Аналогічно відбувається процес створення категорій на сторінці “Categories” (див. рисунок 3.11).



Рисунок 3.11 – Сторінка “Categories” адміністратора

Налаштування дозволяють змінювати статус категорій – активна чи неактивна категорія на стороні клієнта. Додавати назву, мета тайтл, опис, а також змінювати колір категорій, під власні потреби. За всі ці можливості та налаштування для адміністратора на сторінці категорій відповідає файл “AdminFaqsCategory” (див. додаток Б). Досить зручним є те, що плагін надає можливість редагування доданого контенту. Тому, в будь-який момент адміністратор зможе змінити інформацію як щодо конкретного питання так і для конкретної категорії.

Після завершення всіх налаштувань адміністратору необхідно зберегти додані зміни перевірити роботу плагіна на фронтенді свого інтернет-магазину. Також необхідно переконатися, що всі функції та додані налаштування та працюють коректно, і користувачі можуть легко знаходити відповіді на свої запитання.

Таким чином, процес налаштування та конфігурації плагіна “Часті питання” включає створення та редагування питань, управління категоріями, налаштування відображення та функцій пошуку, а також підтримку багатомовності та зворотного зв'язку, який реалізується за допомогою кнопки “Задати питання”. Всі ці дії надають ефективну інтеграцію плагіна на платформу PrestaShop та покращують взаємодію користувачів з сайтом.

3.4 Приклади використання плагіна “Часті питання”

Розроблений плагін “Часті питання” може використовуватися у інтернет-магазинах різної тематики та цільової аудиторії. Необхідно лише налаштувати плагін під свої потреби, додати власні часті запитання до відповіді до них, а також налаштувати дизайн плагіна відповідно до своєї теми.

Варто зазначити, що використання плагіна “Часті питання” на базі CMS PrestaShop значно вплине та покращить взаємодію з клієнтами. Так як плагін забезпечить швидку відповідь на питання, які будуть розміщені на інтернет-магазині у вкладці чстих питань та клієнт злегкістю зможе знайти відповідь на своє запитання.

3.4.1 Приклад використання плагіна в інтернет-магазині

Прикладів використання плагіна “Часті питання” може бути безліч, так як кожен інтернет-магазин створює питання та використовує плагін в залежності від своїх потреб. На прикладі інтернет-магазину одягу було реалізовано приклад використання та можливостей плагіна “Часті питання”.

Першим і надважливим етапом є створення частого питання та відповіді. Необхідно перейти в адмінпанелі у вкладку “FAQs” обравши “Questions/Answers”. Для прикладу створимо питання, що стосується способів оплати замовлення (див. рисунок 3.12).

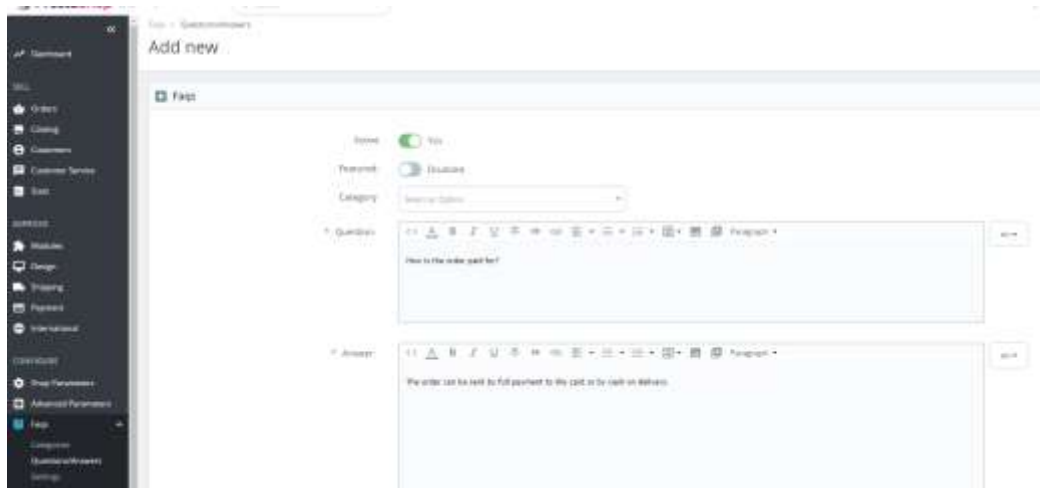


Рисунок 3.12 – Створення питання та відповіді

При створенні запитання було обрано, щоб питання було активне, додано питання та відповідь на нього. Також після створення категорій чатих питань у полі “Category” можна прив’язати дане питання до конкретної категорії. Нажавши кнопку “Save”, зберігаємо створене питання.

Після зберігання питання має з’явитися в списку як зображено на рисунку 3.13.

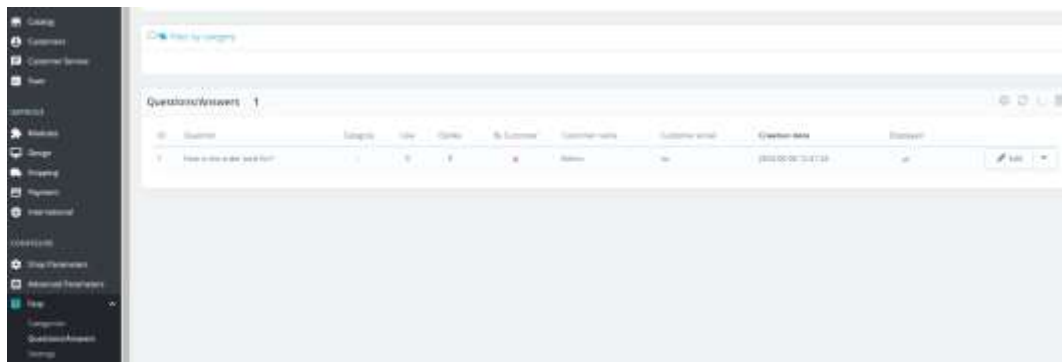


Рисунок 3.13 – Відображення частого питання в списку

За аналогією створюються і категорії для частих питань. Перейшовши у вкладку “Categories” було створено категорію частих питань “Payment for the order”, з якою і пов’яжемо створене часте питання. Створення категорії “Payment for the order” зображено на рисунку 3.14.

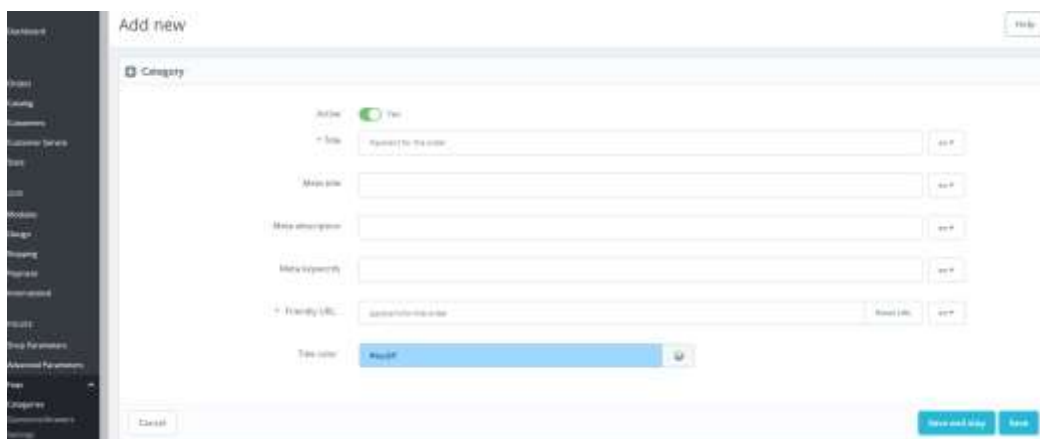


Рисунок 3.14 – Створення категорії “Payment for the order”

При створенні категорії частих питань адміністратор може активувати категорію, змінювати колір категорії, а також додавати мета опис, назву та ключі, завдяки яким пошукові системи швидше зможуть знайти інформацію.

Створивши питання та категорію в адміністративній панелі платформи, було перейдено до перевірки коректності відображення плагіна на магазині, збоку користувача.

На рисунку 3.15 зображено сторінку “Часті питання” з доданою категорією та питанням, яке прив’язане до цієї категорії.

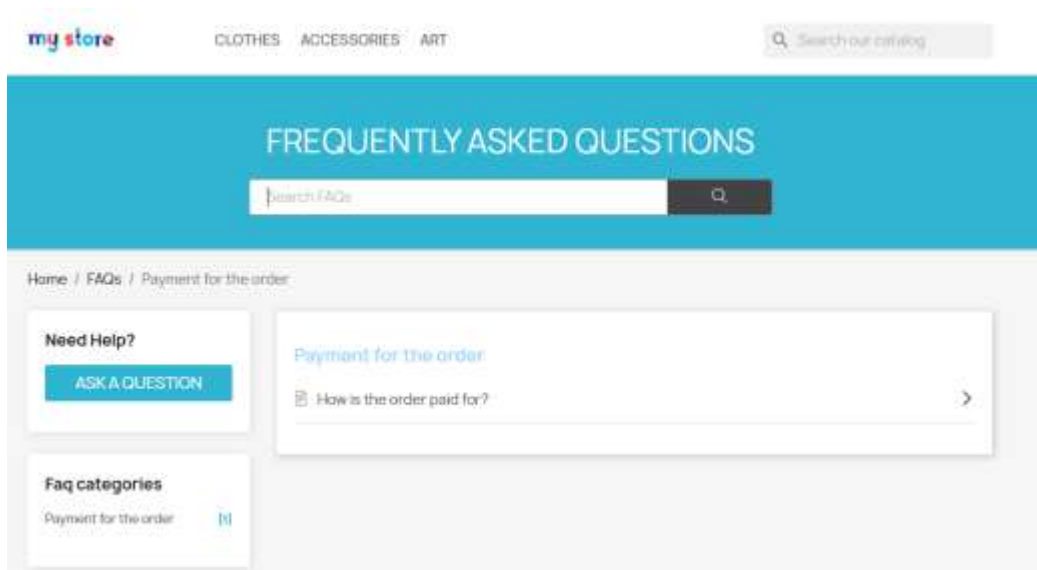


Рисунок 3.15 – Сторінка “Часті питання”

На рисунку 3.16 зображено відображення вкладки “Часті питання” на сторінці категорії продуктів.

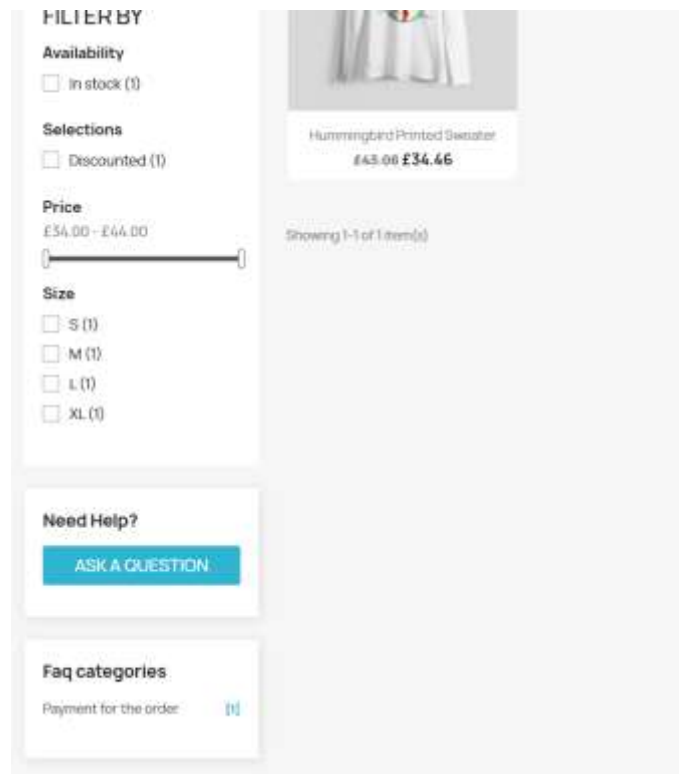


Рисунок 3.16 – Сторінка категорії продуктів із вкладкою “Часті питання”

На даній сторінці зображено вкладку “Часті питання”, по якій користувач може перейти та знайти конкретне питання.

Отже, для прикладу використання плагіна було створено часте питання та відповідну йому категорію. Додавши усі налаштування відповідно відображення на сторінках, позицій та кольору було переглянуто коректність відображення та роботи плагіна, яка виконалася успішно.

3.4.2 Приклад використання форми для задання власного питання

Трапляються випадки, коли все ж клієнт не може знайти відповідь на своє питання у розділі “Часті питання”, тому для цього було створено форму “Задати питання”, яка допоможе користувачеві отримати від продавця відповідь на питання, яке його цікавить. Біля категорій товару на магазині розміщена кнопка

для задання власного питання, нажавши на ню користувач зможемо задати питання. (див. рисунок 3.17).

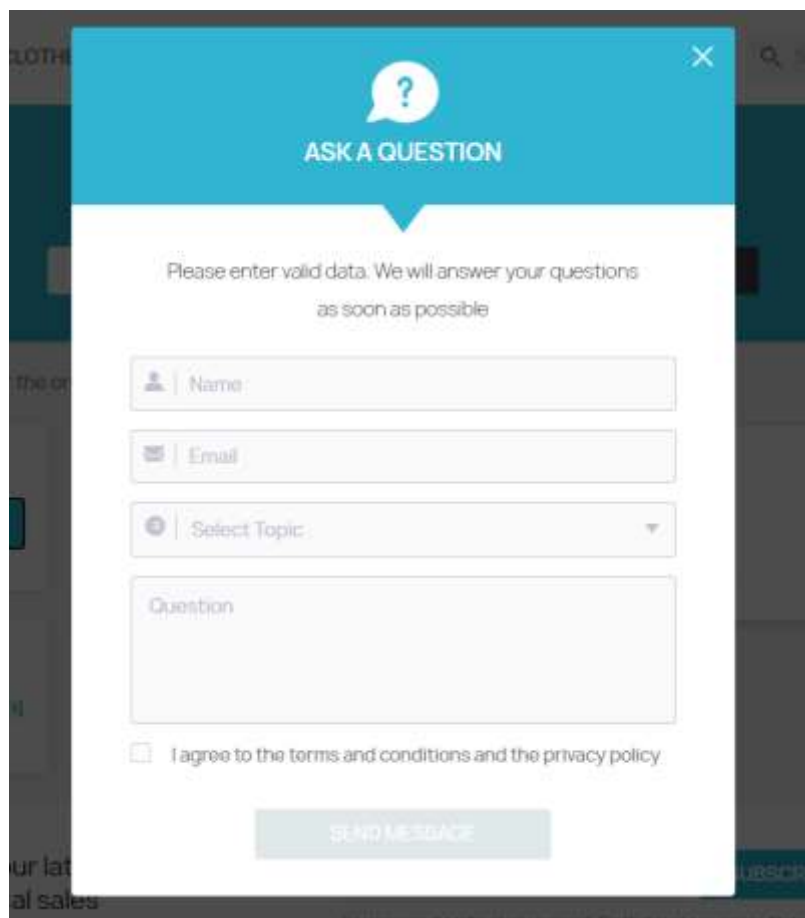


Рисунок 3.17 – Відображення форми “Ask a question”

При заданні питання користувач надає свою пошту, яка допоможе швидко сконтактувати адміністратору магазину із користувачем та відповісти на питання. Забезпечує основні функції в роботі форми файл “AjaxForm.php” (див. додаток В). За допомогою даного файл відбувається отримання та передача даних, які надає користувач у формі. Після чого відбувається перевірка коректності введеної інформації з полів “Name”, “Email” та “Question”. Якщо всі поля є валідними, то вони зберігаються в базу даних та після цього формується надсилання сповіщення адміністратору про надходження нового питання заданого з форми.

3.5 Висновок до третього розділу

В даному розділі було розглянуто процес розробки плагіна “Часті питання” для інтернет-магазину на базі CMS PrestaShop. Було описано процес підготовки середовища для розробки, реалізація основних можливостей плагіна та відображення функціоналу плагіна та його роботи.

Наступним етапом було оглянуто функціонал CMS платформи PrestaShop, що надало змогу розуміти структуру платформи та основні можливості платформи. Після чого було відображено процес встановлення плагіна на магазин. Після успішного завантаження плагіна на платформу, огляду функціоналу було наведено приклади використання плагіна “Часті питання” в інтернет-магазині, а саме створення питання, створення категорії частих питань, при’язка питання до конкретної категорії та відображення даних налаштувань в інтернет-магазині.

Підсумовуючи весь аналіз та огляд функціоналу слід зазначити, що розробка та використання плагіна “Часті питання” надасть змогу інтрнет-магазинам покращити свою функціональність, зручність та пришвидшить взаємодію з користувачами.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Роль центральної нервової системи в трудовій діяльності людини

Центральна нервова система відіграє ключову роль у забезпеченні ефективної та безпечної трудової діяльності людини. Вона є основним регулятором всіх фізіологічних процесів, відповідальних за координацію рухів, підтримання уваги, прийняття рішень та адаптацію до змінних умов праці. Функціонування центральної нервової системи визначає здатність людини виконувати складні завдання, реагувати на стресові ситуації та забезпечувати високу продуктивність праці[30].

Однією з основних функцій центральної нервової системи є обробка інформації, що надходить від сенсорних органів. Під час виконання трудових завдань, людина постійно отримує величезну кількість інформації з зовнішнього середовища. Ця інформація передається в мозок через нервові волокна, де вона аналізується і перетворюється на конкретні дії. Наприклад, при роботі з комп'ютером, очі фіксують зображення на екрані, передають його в мозок, який приймає рішення про наступні дії користувача. Таким чином, центральна нервова система забезпечує точність і швидкість виконання завдань.

Координація рухів є ще однією важливою функцією центральної нервової системи. Завдяки їй, людина може виконувати складні маніпуляції з інструментами, працювати з важким обладнанням або здійснювати точні рухи під час зборки дрібних деталей. Мозок, спинний мозок та периферійні нерви спільно забезпечують гармонійне функціонування м'язів і суглобів, що дозволяє виконувати рухи з необхідною точністю і силою [31]. Спинний мозок отримує сигнали від мозку і передає їх через периферійні нерви до м'язів і суглобів. Він виконує рефлекторні рухи, які реагують на негайні стимули без участі мозку. Крім того, він контролює рухові функції, такі як ходьба, баланс і стабільність. Периферійні нерви, які виходять з мозку і спинного мозку, передають інформацію від сенсорних органів до центральної нервової системи та від

центральної нервової системи до м'язів і суглобів. Ці нерви є носіями інформації, яка дозволяє мозку і спинному мозку взаємодіяти з різними частинами тіла для забезпечення координації рухів.

Центральна нервова система також відповідає за підтримання уваги і концентрації. Під час тривалої роботи, особливо в умовах монотонії, важливо зберігати високий рівень уваги для уникнення помилок та аварійних ситуацій. Мозок постійно обробляє вхідні сигнали і підтримує стан настороженості, що дозволяє вчасно реагувати на зміну обставин. Втома та перевантаження можуть негативно вплинути на функціонування центральної нервової системи, що призводить до зниження продуктивності і підвищення ризику травм.

Реакція на стресові ситуації також регулюється центральною нервовою системою. В умовах високого психоемоційного навантаження, мозок активує захисні механізми, які допомагають організму адаптуватися до стресу. Ця здатність є важливою для роботи в екстремальних умовах або під час виконання відповідальних завдань, де відсутність правильної реакції може призвести до серйозних наслідків. До того ж, умови стресу можуть виникати у різних сферах життя, включаючи робочі ситуації, особисте життя, а також під час виконання навіть звичайних щоденних обов'язків [32]. В такі моменти мозок активує складні механізми реакції на стрес, які регулюються різними частинами центральної нервової системи. Один із ключових механізмів реакції на стрес - це активація симпатичної нервової системи. Ця система мобілізує організм на боротьбу зі стресором, виділяючи нейромедіатори, такі як адреналін і норадреналін. Ці речовини підвищують тонус серцево-судинної системи, підвищують кровообіг, розширюють дихальні шляхи, підвищують концентрацію глюкози в крові, що допомагає організму зреагувати на стресор.

Центральна нервова система відіграє важливу роль у навчанні та професійному розвитку. Здатність до засвоєння нових знань і навичок базується на нейропластичності мозку, тобто його здатності змінювати свою структуру і функції у відповідь на новий досвід. Це дозволяє працівникам постійно

вдосконалювати свої професійні вміння, адаптуватися до нових технологій і методів роботи, що є важливим фактором успішної трудової діяльності.

Таким чином, центральна нервова система є фундаментальним елементом, який забезпечує ефективну та безпечну трудову діяльність людини. Вона регулює обробку інформації, координацію рухів, підтримання уваги, реакцію на стрес та процеси навчання. Підтримання здорового функціонування ЦНС через правильний режим праці, відпочинок, здорове харчування та інші фактори є важливим для забезпечення високої продуктивності та безпеки на робочому місці.

4.2 Перша допомога людині, яка уражена електричним струмом

Надання першої допомоги людині, ураженій електричним струмом, є критично важливим для збереження її життя та здоров'я. Електричний струм може спричинити серйозні травми, включаючи зупинку серця, опіки та пошкодження внутрішніх органів. Тому правильні та своєчасні дії можуть мати вирішальне значення.

Перш за все, необхідно забезпечити безпеку рятувальника. Не можна торкатися ураженої людини, доки вона перебуває під впливом струму. Слід негайно вимкнути джерело живлення, якщо це можливо. Якщо вимкнення струму не є можливим, потрібно використати сухий, непровідний предмет, такий як дерев'яна палиця або пластмасовий предмет, щоб відштовхнути людину від джерела струму. Рятувальник повинен стояти на сухій, непровідній поверхні, щоб уникнути ураження. Не торкайтеся оголених частин тіла потерпілого, якщо ви самі не впевнені, що не контактуєте з джерелом струму [33].

Після відключення електричного струму необхідно оцінити стан постраждалого. Переконайтеся, що потерпілий дихає і має пульс. Якщо він непритомний, негайно зателефонуйте за номером 103. Якщо він не дихає або не має пульсу, слід негайно розпочати серцево-легеневу реанімацію [34]. Для цього потрібно виконати чергування натискань на грудну клітину та штучного

дихання. Оптимальне співвідношення – 30 натискань на грудну клітину до 2 вдихів. Натискання на грудну клітину слід проводити ритмічно, з частотою приблизно 100-120 натискань на хвилину, глибина натискань повинна бути приблизно 5-6 см. Продовжуйте реанімацію до приїзду швидкої допомоги.

Якщо постраждалий дихає, але перебуває у несвідомому стані, його слід покласти в стабільне бокове положення. Це допоможе запобігти аспірації рідини або блювотних мас і забезпечить прохідність дихальних шляхів [35]. Постраждалого необхідно оглянути на наявність опіків, зокрема в місцях входу і виходу електричного струму. Якщо у потерпілого є опіки, охолодіть їх чистою водою протягом 10-15 хвилин. Не розкривайте пухирі, якщо вони утворилися. Опіки слід накрити стерильними пов'язками або чистою тканиною, щоб запобігти інфікуванню.

Якщо постраждалий перебуває у свідомості, перевірте, чи немає небезпеки від джерела струму, і якщо це можливо, відключіть його. Після цього негайно надайте допомогу. Якщо струм не вимкнуто і неможливо виключити його, не наближайтесь до постраждалого, а шукайте предмет, яким можна відірвати чи відтягти контакт [36].

У всіх випадках ураження електричним струмом слід негайно викликати швидку медичну допомогу, навіть якщо постраждалий здається здоровим. Електричні травми можуть мати відстрочені ефекти, включаючи порушення ритму серця або пошкодження внутрішніх органів, які можуть проявитися пізніше. Медичний персонал зможе провести необхідні обстеження та надати кваліфіковану допомогу.

Психологічна підтримка також важлива, оскільки ураження електричним струмом може спричинити сильний стрес та страх у постраждалого. Спокійні та впевнені дії рятувальника допоможуть зменшити тривогу і підтримати моральний стан потерпілого до прибуття медиків [37].

Таким чином, надання першої допомоги при ураженні електричним струмом включає кілька критичних етапів: забезпечення безпеки рятувальника, відключення постраждалого від джерела струму, оцінка його стану, проведення

серцево-легенева реанімація при необхідності, обробка опіків та виклик медичної допомоги. Правильне та своєчасне виконання цих дій може врятувати життя та зменшити ризик серйозних ускладнень.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи освітнього рівня “Бакалавр” було спроектовано та реалізовано плагін “Часті питання”, який призначений для створення розділу з частими питаннями від користувачів та відповідями на них. Плагін було реалізовано за допомогою засобів PHP, MySQL, HTML, CSS та JavaScript.

У першому розділі кваліфікаційної роботи:

- проведено аналіз актуальності використання плагінів та CMS PrestaShop;

- розглянуто основні функції та можливості CMS PrestaShop;
- визначено особливості розробки плагінів для CMS PrestaShop.

У другому розділі кваліфікаційної роботи:

- описано основні вимоги до плагіна;
- визначено акторів та діаграми варіантів використання;
- спроектовано структуру функціональних компонентів;
- сформовано архітектуру плагіна “Часті питання”;
- створено структуру БД
- спроектовано файлову структуру плагіна “Часті питання”.

У третьому розділі кваліфікаційної роботи:

- реалізовано основні функції плагіна;
- впроваджено плагін в інтернет-магазин;
- описано інструкцію щодо встановлення плагіна;
- відображено функціонал плагіна “Часті питання”;
- наведено основні приклади використання плагіна.

У розділі «Безпека життєдіяльності, основи охорони праці» були розглянуті питання щодо ролі центральної нервової системи в трудовій діяльності людини та першої допомоги людині, яка уражена електричним струмом.

ПЕРЕЛІК ДЖЕРЕЛ

1. Заяць О., Капко Я. СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ. Економіка та суспільство. 2023. № 2023. С. 9. URL: <https://doi.org/10.32782/2524-0072/2023-55-65> (дата звернення: 20.02.2024).
2. Матвієнко О., Гончаренко О. Вдосконалення функціоналу інтернет-магазинів за допомогою плагінів : навч. посіб. Харків : Вісн. Нац. техн. ун-ту «Харків. політехн. ін-т», 2020. 118 с.
3. Кулішова О., Батракова С. Аналіз сучасних тенденцій розробки плагінів для електронної комерції. Київ : Наук. вісн. Укр. акад. наук з проблем інформатики, 2023. 114 с.
4. Воробйов О.В. Проблеми та перспективи розвитку CMS-систем для створення інтернет-магазинів : навч. посіб. 3-тє вид. Харків : Вісн. НТУ «ХПІ», 2020. 116 с.
5. Ковальчук О., Петренко О. 5. Порівняльний аналіз популярних CMS-систем для створення веб-сайтів : підручник. Київ : Вісн. Нац. техн. ун-ту «КПІ», 2019. 112 с.
6. Переваги та недоліки різних CMS. W3Schools. URL: <https://www.w3schools.in/wordpress/introduction-to-content-management-system-cms> (дата звернення: 02.02.2024).
7. Тенденції розвитку CMS в сфері електронної комерції. Gartner. URL: <https://www.gartner.com/reviews/market/digital-commerce> (дата звернення: 01.02.2024).
8. Шевченко О., Петренко О. Сучасні CMS-системи: можливості та перспективи розвитку : навч. посіб. 2-ге вид. Львів : Вісн. Нац. ун-ту «Львів. політехніка», 2021. 118 с.
9. вибір CMS-системи для створення інтернет-магазину: критерії та рекомендації : конспект лекцій. Харків : Вісн. Нац. техн. ун-ту «Харків. політехн. ін-т», 2020. 127 с.

10. PrestaShop Official Documentation. Devdocs. URL: <https://devdocs.prestashop-project.org/> (дата звернення: 03.06.2024).
11. PrestaShop Community Forums. PrestaShop. URL: <https://www.prestashop.com/forums/> (дата звернення: 30.05.2024).
12. PrestaShop Developer Resources. Devdocs. URL: <https://devdocs.prestashop-project.org/> (дата звернення: 20.05.2024).
13. PrestaShop Blog PrestaShop. URL: <https://www.prestashop.com/forums/topic/278867-what-is-the-best-way-to-run-a-blog-in-prestashop/> (дата звернення: 10.01.2024).
14. Розробка модулів для PrestaShop. Nicolas Descombes URL: <https://devdocs.prestashop-project.org/8/modules/> (дата звернення: 17.02.2024).
15. Ковальчук О., Петренко О. Розробка модулів для PrestaShop: теоретичні та практичні аспекти : навч. посіб. Київ : Вісн. Нац. техн. ун-ту «КПІ», 2019. 94 с.
16. What is PHP? Write your first PHP Program. Guru99 URL: <https://www.guru99.com/what-is-php-firstphp-program.html>. (дата звернення: 22.02.2024).
17. PHP: The Complete Reference. Amazon. URL: <https://www.amazon.com/PHP-Complete-Reference-Steven-Holzner/dp/0071508546> (дата звернення: 19.02.2024).
18. JavaScript and JQuery: Interactive Front-End Web Development. Amazon. URL: <https://www.amazon.com/javascript/s?k=javascript> (дата звернення: 22.05.2024).
19. HTML and CSS: Designing and Building Web Amazon. URL: <https://www.amazon.com/html-css/s?k=html+and+css> (дата звернення: 12.04.2024)
20. Гончаренко О., Матвієнко О. Вибір середовища розробки для PHP-проектів: критерії та рекомендації. : навч. посіб. 25-те вид. Харків : Нац. техн. ун-ту «Харків. політехн. ін-т»., 2020. 132 с..
21. Документація PHPStorm. Jetbrain. URL: <https://www.jetbrains.com/phpstorm/resources/> (дата звернення: 06.05.2024).

22. PrestaShop Developer's Handbook Devdocs. URL: <https://devdocs.prestashop-project.org/> (дата звернення: 04.02.2024).
23. Mastering PrestaShop. Addons PrestaShop. URL: <https://addons.prestashop.com/en/art-culture-themes/31468-online-book-and-stationary-store.html> (дата звернення: 20.02.2024).
24. 3-Tier Architecture: A Complete Overview. Jinfonet. URL: <https://www.jinfonet.com/resources/bidefined/3-tier-architecture-complete-overview/> (дата звернення: 21.01.2024).
25. 5 Benefits of a 3-Tier Architecture. Izenda. URL: <https://www.izenda.com/5-benefits-3-tier-architecture/>. (дата звернення: 20.02.2024).
26. ER Diagram: Entity Relationship Diagram Model DBMS Example. Casoo URL: <https://casoo.com/templates/er-database-diagram> (дата звернення: 23.02.2024).
27. What is MySQL? Everything You Need to Know. Talend. URL: <https://www.talend.com/resources/what-ismysql/> (дата звернення: 29.01.2024).
28. Rouse M. MySQL. TechTarget. URL: <https://searchoracle.techtarget.com/definition/MySQL> (дата звернення: 29.01.2024).
29. Site File Structure. Web Style Guide URL: <https://webstyleguide.com/wsg3/5-site-structure/3-site-filestructure.html>. (дата звернення: 29.01.2024).
30. Чортюк Ю. В. Центр економічних досліджень та реформ (ЦЕР): Вплив факторів трудового середовища на стан здоров'я працівників та їх працездатність. library.if.ua. URL: https://essuir.sumdu.edu.ua/bitstream/123456789/28559/1/Ekonomichni_factory.pdf. (дата звернення: 07.06.2024)..
31. Роль центральної нервової системи в трудовій діяльності людини : підручник. Київ : Ін-ту харч. хімії і технології НАН, 2019. 67 с. URL: <https://library.if.ua/book/9/920.html> (дата звернення: 07.06.2024).
32. Овчаренко О. Психологія стресу та стресових розладів. 2-ге вид. Київ : УН-Т «Україна», 2019. 266 с.

33. МОЗ. Як надати першу допомогу при опіках. Міністерство охорони здоров'я України. URL: <https://www.kmu.gov.ua/news/moz-nagadye-yak-nadati-pershu-dopomogu-pri-termichnih-opikah> (дата звернення: 10.06.2024).

34. Перша допомога при ураженні електричним струмом. Департамент соціальних питань та охорони здоров'я Южноукраїнської міської ради. URL: http://www.yu.mk.ua/news/show/persha_dopomoga_pri_urazhenni_elektrichnim_strumom (дата звернення: 10.06.2024).

35. Серцево-легенева реанімація: навички, які можуть врятувати людину. Бережи себе. URL: <https://berezhy-sebe.com/navychky-sertsevo-lehenevoi-reanimatsii/> (дата звернення: 10.06.2024).

36. Л. Гвозденко, В. Каньшин. ДСТУ 7239:2011 "Система стандартів безпеки праці. Засоби індивідуального захисту. Загальні вимоги та класифікація". ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ. URL: https://dnaop.com/html/62679/doc doc-ДСТУ_7239_2011#google_vignette (дата звернення: 10.06.2024).

37. Як врятувати життя людині, яку вразило струмом. URL: <https://oe.if.ua/uk/articles/5d259756db9c421dd70dce51> (дата звернення: 10.06.2024).

ДОДАТКИ

Програмний код файлу AdminFaqPost.php

```

public function renderForm()
{
    $class = '';
    $id_lang = $this->context->language->id;
    $id_shop = Shop::getContext() !== 4 ? $this->context->shop->id :
null;
    $categories = faqsCategory::getCategories($id_lang,$id_shop);
    $obj = new faqsPost(Tools::getValue('id_gomakoil_faq'));

    if(!$obj->association){
        $class = ' hide_form_settings';
    }

    $selected_categories =
faqsPost::getAssociatedProductCategories(Tools::getValue('id_gomak
oil_faq'));
    $selected = array();

    $selected_products =
faqsPost::getAssociatedProducts(Tools::getValue('id_gomakoil_faq'
));
    $selected_prod = array();

    if(isset($selected_products[0]['id_product']) &&
$selected_products[0]['id_product']){
        foreach($selected_products as $val){
            $selected_prod[] = $val['id_product'];
        }
    }
    else{
        $selected_prod = array();
    }

    if(isset($selected_categories[0]['id_category']) &&
$selected_categories[0]['id_category']){
        foreach($selected_categories as $val){
            $selected[] = $val['id_category'];
        }
    }
    else{
        $selected = array();
    }

    $this->fields_form = array(
        'tinymce' => true,
        'legend' => array(
            'title' => $this->l('Faqs'),

```



```

        'icon' => 'icon-plus-sign-alt'
    ),
    'input' => array(
        array(
            'type' => 'switch',
            'label' => $this->l('Active'),
            'name' => 'active',
            'is_bool' => true,
            'values' => array(
                array(
                    'id' => 'display_on',
                    'value' => 1,
                    'label' => $this->l('Yes')),
                array(
                    'id' => 'display_off',
                    'value' => 0,
                    'label' => $this->l('No')),
            ),
        ),
    ),
    array(
        'type' => 'switch',
        'label' => $this->l('Featured:'),
        'name' => 'most',
        'required' => false,
        'values' => array(
            array(
                'id' => 'most_on',
                'value' => 1,
                'label' => $this->l('Enabled')
            ),
            array(
                'id' => 'most_off',
                'value' => 0,
                'label' => $this->l('Disabled')
            )
        )
    ),
),

array(
    'type' => 'select',
    'label' => $this->l('Category:'),
    'name' => 'categoryBox',
    'class' => 'chosen',
    'default_value' => (int)$obj->id_gomakoil_faq_category,
    'form_group_class' => 'mpm-faqs-question-parent-category',
    'options' => array(
        'query' => $categories,
        'id' => 'id_gomakoil_faq_category',
        'name' => 'name',
        'class' => 'name',
        'value' => 'id_gomakoil_faq_category'
    )
),
),

```

```

array(
  'type' => 'textarea',
  'label' => $this->l('Question:'),
  'name' => 'question',
  'form_group_class'=> 'textarea_block question_field',
  'lang' => true,
  'required' => true,
  'autoload_rte' => true,
  'rows' => 10,
  'cols' => 100,
  'hint' => $this->l('Invalid characters:').' <>;=#{'
),
array(
  'type' => 'textarea',
  'label' => $this->l('Answer:'),
  'name' => 'answer',
  'form_group_class'=> 'textarea_block answer_field',
  'autoload_rte' => true,
  'required' => true,
  'lang' => true,
  'rows' => 10,
  'cols' => 100,
  'hint' => $this->l('Invalid characters:').' <>;=#{'
),
array(
  'type' => 'text',
  'label' => $this->l('Meta title'),
  'name' => 'meta_title',
  'lang' => true,
  'class' => 'meta_title',
),
array(
  'type' => 'text',
  'label' => $this->l('Meta description'),
  'name' => 'meta_description',
  'lang' => true,
  'class' => 'meta_description',
),
array(
  'type' => 'text',
  'label' => $this->l('Meta keywords'),
  'name' => 'meta_keywords',
  'lang' => true,
  'class' => 'meta_keywords',
),
array(
  'type' => 'tags',
  'label' => $this->l('Tags:'),
  'name' => 'tags',
  'lang' => true,
  'size' => 48
),
array(

```

```

        'type' => 'text',
        'label' => $this->l('Friendly URL:'),
        'name' => 'link_rewrite',
        'lang' => true,
        'required' => true,
        'hint' => $this->l('Only letters and the minus (-)
character are allowed. '),
        'form_group_class' => 'faqs-new-question-link-rewrite-
form-group',
        'maxlength' => '128',
        'suffix' => ' <buuton class ="btn btn-default reset-link-
rewrite-btn">Reset URL</button>',
    ),
    array(
        'type' => 'hidden',
        'name' => 'token_faq',
    ),
    array(
        'type' => 'hidden',
        'name' => 'idLang',
    ),
    array(
        'type' => 'hidden',
        'name' => 'idShop',
    ),
    array(
        'type' => 'switch',
        'label' => $this->l('Category/Product association'),
        'name' => 'association',
        'required' => false,
        'values' => array(
            array(
                'id' => 'association_on',
                'value' => 1,
                'label' => $this->l('Enabled')
            ),
            array(
                'id' => 'association_off',
                'value' => 0,
                'label' => $this->l('Disabled')
            )
        )
    ),
),
array(
    'type' => 'html',
    'name' => 'html_data',
    'form_group_class' => 'block_line_after_form
block_more_settings'.$class,
    'html_content' => '<div
class="line_after_form"></div>',
),

```

```

array(
    'type' => 'switch',
    'label' => $this->l('Hide question in faq page'),
    'name' => 'hide_faq',
    'form_group_class'=> 'block_more_settings'.$class,
    'required' => false,
    'values' => array(
        array(
            'id' => 'hide_faq_on',
            'value' => 1,
            'label' => $this->l('Enabled')
        ),
        array(
            'id' => 'hide_faq_off',
            'value' => 0,
            'label' => $this->l('Disabled')
        )
    )
),

array(
    'type' => 'categories',
    'label' => $this->l('Category association'),
    'name' => 'categoriesBox',
    'form_group_class'=> 'block_more_settings'.$class,
    'tree' => array(
        'use_search' => false,
        'id' => 'categoriesBox',
        'use_checkbox' => true,
        'selected_categories' => $selected,
    ),
    'values' => array(
        'trads' => array(
            'Root' => Category::getTopCategory(),
            'selected' => $this->l('Selected'),
            'Collapse All' => $this->l('Collapse All'),
            'Expand All' => $this->l('Expand All'),
            'Check All' => $this->l('Check All'),
            'Uncheck All' => $this->l('Uncheck All')
        ),
        'selected_cat' => $selected,
        'input_name' => 'categoryBox[]',
        'use_radio' => false,
        'use_search' => false,
        'disabled_categories' => array(),
        'top_category' => Category::getTopCategory(),
        'use_context' => true,
    )
),

array(

```

```

        'type' => 'html',
        'label' => $this->l('Product association'),
        'name' => 'html_data',
        'form_group_class'=> 'block_select_products
block_more_settings'.$class,
        'html_content' => $this-
>getBlockSearchProduct($selected_prod),
    ),

    array(
        'type' => 'html',
        'name' => 'html_data',
        'form_group_class'=> 'block_line_after_form
block_more_settings'.$class,
        'html_content' => '<div
class="line_after_form"></div>',
    ),

    array(
        'type' => 'switch',
        'label' => $this->l('Open as a url:'),
        'name' => 'as_url',
        'required' => false,
        'values' => array(
            array(
                'id' => 'as_url_on',
                'value' => 1,
                'label' => $this->l('Enabled')
            ),
            array(
                'id' => 'as_url_off',
                'value' => 0,
                'label' => $this->l('Disabled')
            )
        )
    ),
),

array(
    'type' => 'switch',
    'label' => $this->l('By customer'),
    'name' => 'by_customer',
    'is_bool' => true,
    'form_group_class' => 'customer-question-btn-fg',
    'values' => array(
        array(
            'id' => 'display_on',
            'value' => 1,
            'label' => $this->l('Yes')),
        array(
            'id' => 'display_off',
            'value' => 0,
            'label' => $this->l('No')),
    ),
),

```

```

    ),
  ),
  array(
    'type' => 'text',
    'label' => $this->l('Customer name'),
    'name' => 'name',
    'form_group_class'=> 'input_disabled_faqs',
    'class' => 'name',

  ),
  array(
    'type' => 'text',
    'label' => $this->l('Customer email'),
    'name' => 'email',
    'class' => 'email',
    'form_group_class'=> 'input_disabled_faqs',
  ),
),
'buttons' => array(
  'save-and-stay' => array(
    'title' => $this->l('Save and stay'),
    'name' => 'submitAdd'.$this->table.'AndStay',
    'type' => 'submit',
    'class' => 'btn btn-default pull-right',
    'icon' => 'process-icon-save'
  )
),
'submit' => array(
  'title' => $this->l('Save'),
)
);

$this->tpl_form_vars['PS_ALLOW_ACCENTED_CHARS_URL'] =
(int)Configuration::get('PS_ALLOW_ACCENTED_CHARS_URL');
$this->fields_value['token_faq'] =
Tools::getAdminTokenLite('AdminFaqsPost');
$this->fields_value['idLang'] = Context::getContext()-
>language->id;
$this->fields_value['idShop'] = Context::getContext()->shop->id;

return parent::renderForm();
}

```

Програмний код файлу AdminFaqsCategory.php

```

    <?php
    if (!defined('_PS_VERSION_')) {
        exit;
    }
    require_once(dirname(__FILE__) . '/../../faqs.php');
    require_once(dirname(__FILE__) .
    '/../../classes/faqsCategory.php');

    class AdminFaqsCategoryController extends ModuleAdminController
    {
        private $_objFaqs;

        public function __construct()
        {
            $this->className = 'faqsCategory';
            $this->table = 'gomakoil_faq_category';
            $this->bootstrap = true;
            $this->lang = true;
            $this->edit = true;
            $this->delete = true;

            parent::__construct();

            $this->_objFaqs = new Faqs();
            $this->_position_identifier = 'id_gomakoil_faq_category';
            $this->_defaultOrderBy = 'a!position';
            $this->orderBy = 'position';

            $this->bulk_actions = array(
                'delete' => array(
                    'text' => $this->l('Delete selected'),
                    'icon' => 'icon-trash',
                    'confirm' => $this->l('Delete selected items?')
                )
            );

            $this->_join .= 'LEFT JOIN
            `'. _DB_PREFIX_ .'gomakoil_faq_category_shop` sa ON
            (a.`id_gomakoil_faq_category` = sa.`id_gomakoil_faq_category`)';
            $this->_select .= 'sa.*';
            $this->_where .= 'AND sa.id_shop IN(' . implode(',',
            Shop::getContextListShopID()) . ') GROUP BY
            sa.id_gomakoil_faq_category';

            $this->fields_list = array(
                'id_gomakoil_faq_category' => array(
                    'title' => $this->l('ID'),
                    'filter_key' => 'a!id_gomakoil_faq_category',

```

```

        'width' => 20
    ),
    'name' => array(
        'title' => $this->l('Title'),
        'width' =>100,
        'filter_key' => 'b!name',
        'orderby' => true
    ),
    'active' => array(
        'title' => $this->l('Displayed'),
        'active' => 'status',
        'filter_key' => 'sa!active',
        'type' => 'bool',
        'width' => 70,
        'orderby' => true
    ),
    'position' => array(
        'title' => $this->l('Position'),
        'width' => 40,
        'filter_key' => 'a!position',
        'align' => 'left',
        'position' => 'position'
    ),
    'date_add' => array(
        'title' => $this->l('Creation date'),
        'maxlength' => 190,
        'width' =>100
    ),
);
}

public function init()
{
    parent::init();
}

public function initContent()
{
    $urlFaqs = trim(faqs::getBaseUrl(), ' /');
    $this->tpl_list_vars['urlFaqs'] = $urlFaqs;

    parent::initContent();
}

public function initProcess(){
    parent::initProcess();
}

/**
 * Must be overridden for correct work of multishop feature
 *
 * @param int $id_object
 * @return bool

```



```

*/
protected function updateAssoShop($id_object)
{
    return true;
}

public function initPageHeaderToolbar(){
    if ($this->display == 'view' || $this->display == 'edit')
    {
        $baseUrl = __PS_BASE_URL_SSL__ . __PS_BASE_URI__;
        $page = new faqsCategory(
Tools::getValue('id_gomakoil_faq_category'),
Context::getContext()->language->id );

        if(!faqs::getRewriteSettings()){
            $faqUrl =
$baseUrl.'index.php?fc=module&module=faqs&controller=display&categ
ory='.$page->link_rewrite;
        } else{
            $home_page_id = Configuration::get('FAQS_SEO_HOME_PAGE') ?
Configuration::get('FAQS_SEO_HOME_PAGE') : 'faqs';
            $faqUrl = $baseUrl.$home_page_id.'/' . $page->link_rewrite
. '.html';
        }

        $this->page_header_toolbar_btn['preview'] = array(
            'href' => $faqUrl,
            'desc' => $this->l('Preview', null, null, false),
            'short' => $this->l('Preview', null, null, false),
            'target' => true,
        );
    }
    parent::initPageHeaderToolbar();
}

public function renderForm()
{
    $this->fields_form = array(
        'legend' => array(
            'title' => $this->l('Category'),
            'icon' => 'icon-plus-sign-alt'
        ),
        'input' => array(
            array(
                'type' => 'switch',
                'label' => $this->l('Active'),
                'name' => 'active',
                'shop' => true,
                'is_bool' => true,
                'values' => array(
                    array(
                        'id' => 'display_on',

```

```

        'value' => 1,
        'label' => $this->l('Yes')),
    array(
        'id' => 'display_off',
        'value' => 0,
        'label' => $this->l('No')),
    ),
),
array(
    'type' => 'text',
    'label' => $this->l('Title'),
    'name' => 'name',
    'lang' => true,
    'required' => true,
    'class' => 'name',
    'form_group_class' => 'faqs-new-category-title-form-
group'
),
array(
    'type' => 'text',
    'label' => $this->l('Meta title'),
    'name' => 'meta_title',
    'lang' => true,
    'class' => 'meta_title',
),
array(
    'type' => 'text',
    'label' => $this->l('Meta description'),
    'name' => 'meta_description',
    'lang' => true,
    'class' => 'meta_description',
),
array(
    'type' => 'text',
    'label' => $this->l('Meta keywords'),
    'name' => 'meta_keywords',
    'lang' => true,
    'class' => 'meta_keywords',
),
array(
    'type' => 'text',
    'label' => $this->l('Friendly URL:'),
    'name' => 'link_rewrite',
    'lang' => true,
    'required' => true,
    'hint' => $this->l('Only letters and the minus (-)
character are allowed.'),
    'form_group_class' => 'faqs-new-category-link-rewrite-
form-group',
    'suffix' => '<button class="btn btn-default reset-link-
rewrite-btn">ResetURL</button>',

```

```

    ),
    array(
        'type' => 'hidden',
        'name' => 'PS_ALLOW_ACCENTED_CHARS_URL',
    ),
    array(
        'type' => 'color',
        'label' => $this->l('Title color'),
        'name' => 'color',
        'hint' => $this->l('Choose a color with the color
picker, or enter an HTML color (e.g. "lightblue", "#CC6600").')
    ),
),
'buttons' => array(
    'save-and-stay' => array(
        'title' => $this->l('Save and stay'),
        'name' => 'submitAdd'.$this->table.'AndStay',
        'type' => 'submit',
        'class' => 'btn btn-default pull-right',
        'icon' => 'process-icon-save'
    )
),
'submit' => array(
    'title' => $this->l('Save'),
)
);

$this->fields_value['PS_ALLOW_ACCENTED_CHARS_URL'] =
(int)Configuration::get('PS_ALLOW_ACCENTED_CHARS_URL');

return parent::renderForm();
}

public function renderList()
{
    $this->addRowAction('edit');
    $this->addRowAction('delete');
    return parent::renderList();
}

public function ajaxProcessUpdatePositions()
{
    $category = Tools::getValue('gomakoil_faq_category');
    foreach($category as $key => $value){
        $value = explode('_', $value);
        Db::getInstance()->update('gomakoil_faq_category',
array('position' => (int)$key,
'id_gomakoil_faq_category='.(int)$value[2]));
    }
}
}
}

```

Програмний код файлу AjaxForm.php

```

        public function displayAjax()
    {
        $json = array();

        try {

            if (Tools::getValue('action') == 'setUsefulness') {
                $json = $this->updateUsefulness();
                $json['success'] = true;
            }

            if (Tools::getValue('action') == 'send') {
                $name = Tools::getValue('name');
                $email = Tools::getValue('email');
                $question_category_id = Tools::getValue('category');
                $question = Tools::getValue('question');

                if (!$name) {
                    throw new Exception ('name_customer:' .
Module::getInstanceByName('faqs')->l('Name is required',
'AjaxForm'));
                }

                if (!$email) {
                    throw new Exception ('email_customer:' .
Module::getInstanceByName('faqs')->l('E-mail is required',
'AjaxForm'));
                } elseif ($email && !Validate::isEmail($email)) {
                    throw new Exception ('email_customer:' .
Module::getInstanceByName('faqs')->l('E-mail is not valid',
'AjaxForm'));
                }

                if (!$question) {
                    throw new Exception ('question_customer:' .
Module::getInstanceByName('faqs')->l('Question is required',
'AjaxForm'));
                }

                $json['form'] = 'response';
                $question_is_saved = $this->saveCustomerQuestion($name,
$email, $question_category_id, $question);

                if (!$question_is_saved) {
                    throw new Exception('Can not save question!');
                }

                $notification_template =

```

```

FaqMailer::getNewQuestionTemplate($this-
>getEmailNotificationVars($question_is_saved));
    $notification_is_sent =
FaqMailer::sendNewQuestionNotificationToAdmins($notification_temp
late);

    if ($question_is_saved && $notification_is_sent) {
        $json['form'] = $this-
>notificationForm(Module::getInstanceByName('faqs')->l('Please,
check your mailbox from time to time. We will respond you as soon
as possible.', 'AjaxForm'), 'success');
    } else {
        $json['form'] = $this-
>notificationForm(Module::getInstanceByName('faqs')->l('Something
went wrong... Can you try again?', 'AjaxForm'), 'error');
    }
}

    if (Tools::getValue('action') == 'getSearchPreview') {
        $json['template'] = $this->getSearchPreview();
    }

    if (Tools::getValue('action') == 'showForm') {
        $json['form'] = $this-
>getFaqForm(Tools::getValue('id_lang'),
Tools::getValue('id_shop'));
    }

    die(json_encode($json));
} catch (Exception $e) {
    $error_info = explode(':', $e->getMessage());
    $json['error_field'] = $error_info[0];
    $json['error_message'] = $error_info[1];

    if ($e->getCode() == 10) {
        $json['error_message'] = $e->getMessage();
    }
}

die(json_encode($json));
}

public function updateUsefulness()
{
    $id_faq = Tools::getValue('id_faq');
    $usefulness_cookies = Tools::getValue('usefulness_cookies');
    $usefulness = Tools::getValue('usefulness');

    $likes = (int)Db::getInstance()->getValue("SELECT `like` FROM
`"._DB_PREFIX_."gomakoil_faq` WHERE `id_gomakoil_faq` =
".(int)$id_faq);
    $dislikes = (int)Db::getInstance()->getValue("SELECT
`dislike` FROM `"._DB_PREFIX_."gomakoil_faq` WHERE

```

```

`id_gomakoil_faq` = "(int)$id_faq);
    if ($usefulness === 'like') {
        $likes++;
    } elseif ($usefulness === 'dislike') {
        $dislikes++;
    }

    if ($usefulness_cookies === 'like') {
        $likes--;
    } elseif ($usefulness_cookies === 'dislike') {
        $dislikes--;
    }

    if ($likes < 0) {
        $likes = 0;
    }
    if ($dislikes < 0) {
        $dislikes = 0;
    }

    Db::getInstance()->execute("UPDATE
`"._DB_PREFIX_"gomakoil_faq` SET `like` = "(int)$likes.",
`dislike` = "(int)$dislikes." WHERE `id_gomakoil_faq` =
"(int)$id_faq);
    return [
        'likes' => $likes,
        'dislikes' => $dislikes
    ];
}

```