Зміст

[Зміст 1](#_Toc516001006)

[Вступ 5](#_Toc516001007)

[1 Опис ЗАСОБІВ РЕАЛІЗАЦІЇ 7](#_Toc516001008)

[1.1 MVC 7](#_Toc516001009)

[1.2 JavaScript 8](#_Toc516001010)

[1.3 HTML 9](#_Toc516001011)

[1.4 CSS 10](#_Toc516001012)

[1.5 SQL 11](#_Toc516001013)

[1.6 PHP 12](#_Toc516001014)

[2 ЕТАПИ РОЗРОБКИ 13](#_Toc516001015)

[2.1 Верстка макету веб-сайту 13](#_Toc516001016)

[2.2 Проектування бази даних 13](#_Toc516001017)

[2.3 Побудова серверної частини 18](#_Toc516001018)

[3 Опис ЗАСОБІВ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ 20](#_Toc516001019)

[3.1 Перевірка введених даних 20](#_Toc516001020)

[3.2 Корзина 21](#_Toc516001021)

[3.3 Панель адміністратора 21](#_Toc516001022)

[Висновки 22](#_Toc516001023)

[Список використаних джерел 23](#_Toc516001024)

[Додаток 24](#_Toc516001025)

Вступ

Наразі важко уявити життя людей без інформаційних, зокрема Web, технологій. Інтернет – це практично нескінченний інформаційний ресурс, він є всесвітнім джерелом знань, глобальним і загальнодоступним по своїй суті. Стрімкий розвиток веб-ресурсів замінив стандартне поняття терміну «Інтернет». Найчастіше під цим словом розуміють веб-простір і доступну через нього інформацію, а не фізичну мережу, що призводить до різноманітних юридичних колізій та правових наслідків.

Мабуть не існує сьогодні компанії, яка б не комунікувала з замовниками чи покупцями через веб-застосунки. Автоматизовані системи дозволяють спростити процес обміну інформації, а отже й зекономити час та збільшити прибуток.

Інтернет-магазин (також називають «електронний магазин», «мережевий магазин», «e-shop», «internet-shop») - це інтерактивний сайт з каталогом, в якому представляються товари і послуги, а також кошиком для формування замовлення.

У правильному інтернет-магазині обов'язково повинні бути присутніми: рекламовані товари і послуги, прайс-лист, контактна інформація (телефон, факс, поштову адресу, e-mail тощо), пропозиції різних варіантів оплати, можливість здійснення покупок.

Робота інтернет-магазину схожа на роботу простого магазину. Клієнт, переглядаючи каталог, поміщає інформацію, що цікавить його позицію в кошик. Для оплати замовлених товарів і послуг пропонуються різні варінат способів оплати.

За допомогою логіна і пароля клієнт має можливість зареєструватися і надалі використовувати свої дані при вході. Інформація про нього зберігається в базі даних магазину. Клієнту також забезпечується конфіденційність здійснюють операції.

Сьогодні, інтернет-магазини стають одним з найбільш перспективних способів ведення і розвитку великого і малого бізнесу при зниженні витрат і збільшенні прибутку

Система в першу чергу повинна бути надійною та мати зручний, графічний інтерфейс, який буде зрозумілим та легким у користуванні для широкого кола користувачів. Розроблюваний застосунок також повинен забезпечувати адмін- частину для редагування та контролю.

Отже, метою даної курсової роботи є створення веб-застосунку, який автоматизує перегляд та замовлення товарів у інтернет магазині, розроблюваний програмний продукт повинен контролювати дані, що надходять зовні та взаємодіяти з користувачами.

Під час розробки програми використовувались стандартні методи наукових досліджень: аналіз, порівняння, прогнози, перевірка прогнозів, індукція, синтез, тестування.

# Опис ЗАСОБІВ РЕАЛІЗАЦІЇ

Для розробки Web-додатків використовуються багато засобів, але відповідно до навчального курсу, для реалізації програмного продукту використовувались наступні технології: мови програмування PHP, JavaScript стандартна мова розмітки HTML5, мова для опису зовнішнього вигляду CSS3, система керувань базами даних MySQL. Продукт побудований на принципах патерна (шаблона) проектування MVC.

## MVC

MVC— архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

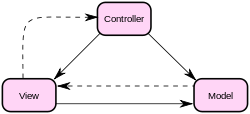


Рисунок 1.1 – Принцип роботи шаблону MVC

Опис компонентів:

* Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.
* Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.
* Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

## JavaScript

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

* Мова JavaScript використовується для:
* написання сценаріїв веб-сторінок для надання їм інтерактивності;
* створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);
* програмування на стороні сервера (Node.js);
* стаціонарних застосунків (Electron, NW.js);
* мобільних застосунків (React Native, Cordova);
* сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
* всередині PDF-документів тощо.

## HTML

HTML — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

* створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
* отримання інформації із Всесвітньої мережі через гіперпосилання;
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

## CSS

**CSS** використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного **CSS**. Стилі для відображення сторінки можуть бути:

* Стилі автора (інформація надана автором сторінки):
  + зовнішні таблиці стилів (англ. *stylesheet*), найчастіше окремий файл або файли **.css**
  + внутрішні таблиці стилів, включені як частина документу або блоку
  + стилі для окремого елементу
* Стилі користувача
  + локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera)
* Стилі переглядача (браузера)
  + стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип *каскадності*, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

## SQL

SQL — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування(C або Pascal), SQL може формувати інтерактивні запити або, бувши вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

## PHP

PHP -  скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

# ЕТАПИ РОЗРОБКИ

## Верстка макету веб-сайту

**2.1.1 Інтерфейс**

Інтерфейс відіграє важливу роль в веб-системі. Адже це те що бачить користувач при знаходженні на сайті, перш за все він звертає увагу на зручність і зрозумілість навігації, кольорову гамму. Від враження, яке справляє зовнішній вигляд вашого сайту, залежить рішення користувача залишитися на сторінці чи ні. Це важливо адже зовнішній вигляд сайту це імідж компанії яка надає послугу.

Веб-інтерфейс вирішує такі важливі завдання, як функціональність, зрозумілість у використанні. Відвідувач сайту повинен легко орієнтуватися на його сторінках, знаходити потрібну інформацію, мати можливість швидко повернутися на головну сторінку.

Отже, для того, щоб не втратити користувачів, веб-сайт повинен бути адаптивним та кросбраузерним.

**2.1.2 Адаптивність**

Для адаптивності було використано такий функціонал CSS, як медіа запити.

Структура цієї всластивості така, що блок починається з ключового слова @media, а далі іде умова при якій код в середені блока буде спрацьовувати. Наприклад: @media screen and (min -width: 1280px) застосує CSS стилі які всередені цього блоку для пристроїв з шириною дисплея більшою чи рівною 1280px. Тобто в залежності від розміру дисплея будуть застосовуватися визначені стилі.

## Проектування бази даних

База даних – це основа будь-якого проекту який має на меті працювати з даними. База яка спроектована по правилам дозволяє зменшити навантаження на SQL сервер, який в свою чергу дозволить зменшити час очікування обробки даних від користувача. Існують правила проектування бази які мають назву нормальні форми. Існує три основні:

* Перша форма
  + Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.
  + Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.
  + Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.
* Друга форма
  + Схема бази даних повинна відповідати вимогам першої нормальної форми.
  + Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.
* Третя форма
  + Схема бази даних повинна відповідати всім вимогам другої нормальної форми.
  + Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

**2.2.1 Фізична модель**

Фізична модель - подання дизайну даних як реалізованого чи призначеного для реалізації у системі керування базами даних. Кожна таблиця має первичний ключ що дозволяє унікально ідентифікувати записи в таблиці. Також за допомогою правил нормалізації було усунено надмірність даних, що могло б привести до помилок при роботі з базою або до високого навантаження сервера. Використовуються зовнішні ключі. Отримана модель задовольняє трьом нормальним формам.

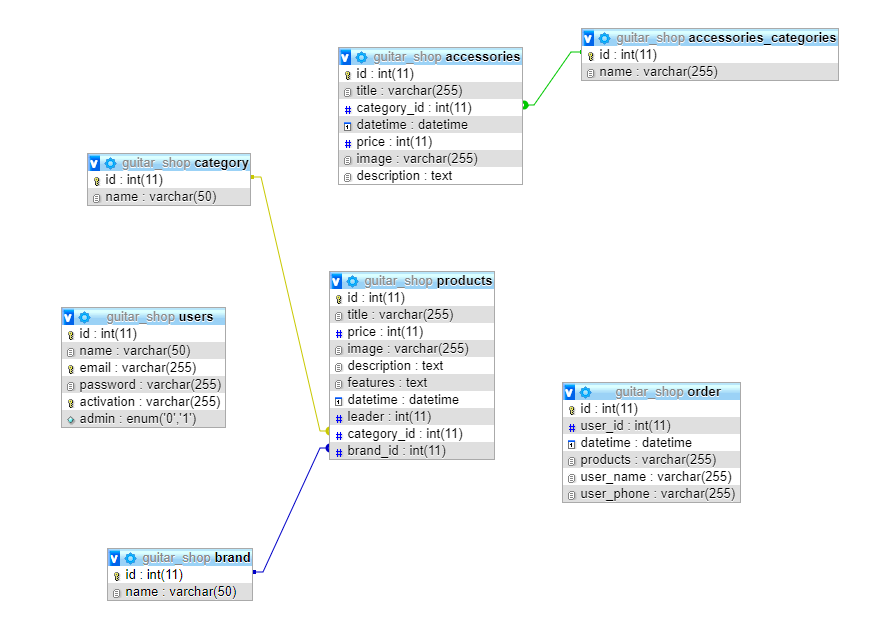


Рисунок 2.1– Фізична модель бази даних

**2.2.2 Опис таблиць**

База даних складається з шести таблиць, в кожній з них, головним ключем є цілочисельне автоінкрементне поле «id».

Таблиця «category» містить назву категорії та її id.

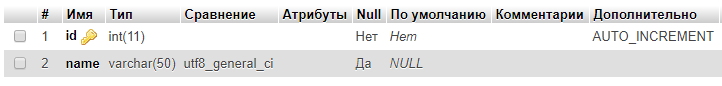


Рисунок 2.2 – Структура таблиці категорій

Таблиця категорій має зв'язок один до багатьох з таблицею products, оскільки продукт може мати лише одну категорію.

В таблиці «brand» зберігаються дані про виробника товару.

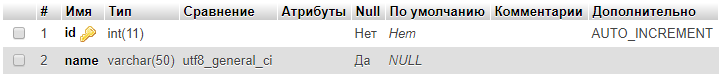


Рисунок 2.3 – Структура таблиці виробників

Таблиця «products» містить інформацію про товари: назву, ціну, шлях до зображення, опис, переваги товару, дата додання товару до бази, чи являється він лідером продаж, «id» категорії та «id» виробника. Таблиця пов’язана із категоріями з допомогою зовнішнього ключа category\_id, та виробниками з допомогою зовнішнього ключа brand\_id.



Рисунок 2.4 – Структура таблиці товарів

Дані про клієнта зберігаються в таблиці «users». Вона містить ім’я користувача, електронну пошту, пароль, ключ активації акаунта за допомогою підтвердження через електронну пошту. Також ця таблиця має інформацію про відношення до адміністративної частини сайту.



Рисунок 2.5 – Структура таблиці користувачів

Таблиця «order» містить інформацію про зроблені замовлення і має наступні поля: «id» користувача який зробив замовлення, його ім’я та номер телефону, дата замовлення, а також індифікатор користувача, якщо він авторизований на сайті береться його ідентифікаційний номер з таблиці користувачів, якщо ж ні, то його ідентифікаційний номер вираховується за допомогою алгоритму . У таблиці присутнє спеціальне поле типу JSON, в якому ключем є індифікатор товару, а значенням його кількість.



Рисунок 2.7 – Структура таблиці замовлень

## Побудова серверної частини

Основою курсового проекту є паттерн проектування MVC. Головною частиною проекту є маршрутизатор. На нього приходить запит з будь-якого адресу і роутер у свої чергу проходить по масиву адресів, находить співпадаючий патерн і викликає ту функцію з якою цей патерн зв’язаний. В мові програмування PHP існує можливість працювати з назвами класів та функції маючи їх представлення у вигляді рядка. У випадку не знаходження потрібного патерна роутер поверне код 404(помилка «не знайдено того, що запросив клієнт») і браузер покаже сторінку помилки.

**2.3.1 Модель**

Основою моделі є SqlBuilder через який будуються запити до таблиць. Також у кожному класі, що симулює таблицю з бази даних сформовано деякі методи, що відповідають за бізнес-логіку, кожен з яких є безпечним, всі дані які приходять від користувача проходять валідацію і це дає змогу зберегти систему від атак злочинців. Клас SqlBuilder який є обгорткою над звичайним SQL запитом. При роботі з даними які приходять від користувача потрібно враховувати безпечність цих даних, оскільки користувач може виконати SQL-ін’єкцій. Для цього використано «PDO», зокрема «prepared statements».

**2.3.2 Представлення**

За представлення даних у проекті відповідає функція output яка приймає назву шаблону та дані для відображення. Дана функція читає шаблон з спеціальної папки, яка має назву templates, далі парсить його і заміняє спеціальні символи на PHP символи. Далі виводиться у браузер. Також відпарсений шаблон зберігається у спеціальну папку cache. Це зроблено для оптимізації, коли браузер запитує шаблон функція перевіряє чи не змінився шаблон. У разі зміни чи відсутності готового до представлення шаблону функція заново парсить і зберігає файл, у іншому випадку віддається файл з папки cache.

**2.3.3 Контролер**

Контролер зв’язує модель і представлення, а також працює з HTTP протоколом. Контролери повинні бути максимально тонкими і не виконувати ніякої бізнес логікі, яка має бути винесена у відповідні моделі. Контролер опрацьовують GET та POST запити, працюють з куками та сесією. Кожен такий клас повинен повернути текст або результат функції представлення.

# Опис ЗАСОБІВ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ

Взаємодія з користувачем повинна бути швидкою, надійною, зрозумілою користувачу. Водночас необхідо ретельно перевіряти отримані дані, щоб уникнути пошкодження системи. Інформація користувачу надається як у вигляді тексту власне на сторінці, так і у вигляді випливаючих вікон з використанням скриптів JavaScript.

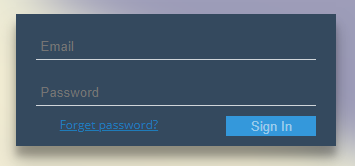


Рисунок 3.1 – Вікно входу в обліковий запис

Коли користувач увійшов до аккаунту він перенаправляється до свого кабінету в якому є посилання на змінення даних та вихід з аккаунту.

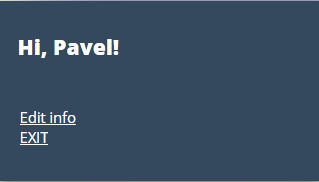


Рисунок 3.2 – Вікно кабінету

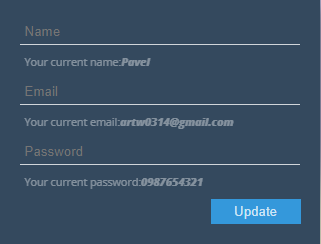


Рисунок 3.3 – Вікно оновлення інформації користувача

## Перевірка введених даних

Перевірка введених даних – важливий аспект будь-якого успішного програмного продукту.

Перевірка введених даних відбувається як браузером так і на стороні сервера. В курсовому проекті мною реалізовані наступні перевірки на стороні клієнта:

* Заповненість полів.
* Коректний формат електронної пошти.
* Ідентичність введених паролів при реєстрації.

## Корзина

Веб-система побудована так, шо користувач на сторінці товару може тільки додовати до корзини існуючий товар, а купівлю проводити тільки в корзині. На сторінці корзини відображуються інвормація про доданий товар, суму, а також кнопка для видалення товари з корзини. У нижній частині корзини розташована загальна сума і кнопка купівлі. При натисканні на цю кнопку система перевіряє чи користувач увійшов на сайт, у задовільному випадку йому пропонують заповнити поле з номером телефону потім замовлення оформляється і використовуються дані про користувача, які були введенні при реєстрації. У разі коли користувач не увійшов, він перенаправляється на форму для заповнення даних для замовнення.

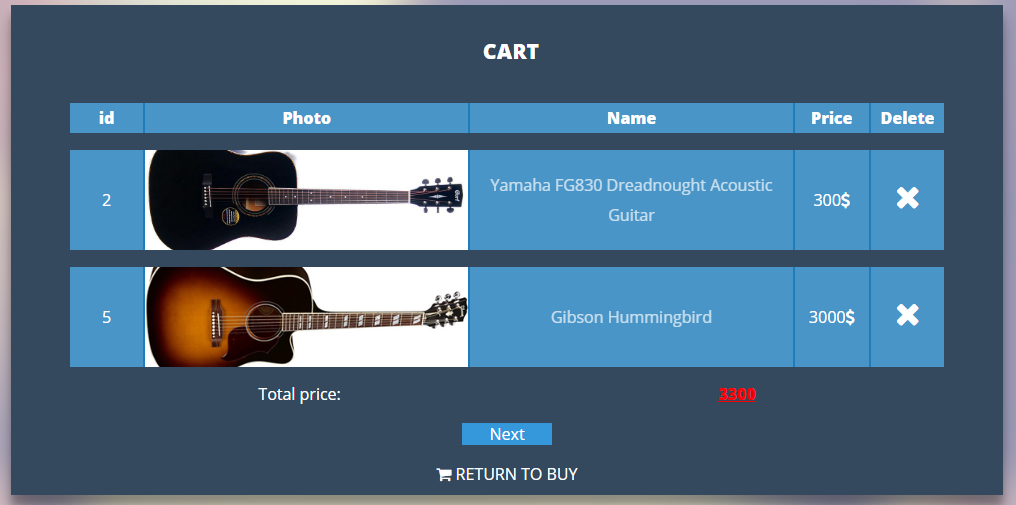


Рисунок 3.4 – Вигляд корзини

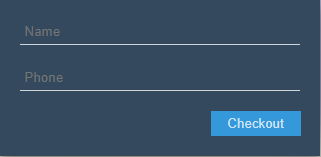


Рисунок 3.5 – Форма для введення даних під час замовлення товару

## Панель адміністратора

Панель адміністратора дає змогу користувачу працювати з даними не використовуючи SQL. Це дає змогу власнику магазину швидко навчити людину для роботи з адмін-частиною сайта.

Також дотримано всіх правил безпеки. Будь-хто не зможе зайти на будь-яку адмін-частину сайту не маючи прав доступу. Права доступу відображені в таблиці users в окремому полі. У разі наявності прав користувач спостерігає нові посилання в його кабінеті, які дозволяють оновити та додати записи до категорій та товарів.



Рисунок 3.6 – Кабінет адміністратора

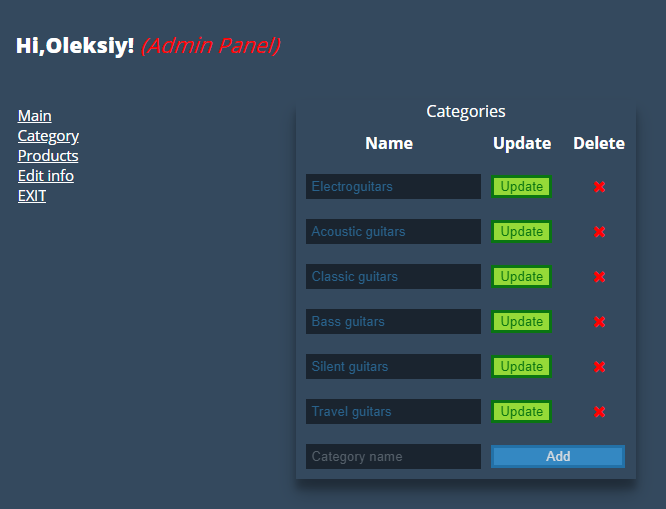


Рисунок 3.7 – Форма змінення, видалення та додавання категорій

# Висновки

В ході курсової роботи було спроектовано та реалізовано програмний продукт відповідно до заданої предметної області «Інтернет-магазин Gitar Keren». Система дозволяє переглядати доступні товари, додавати в корзину, робити замовлення, сортувати товар. Також реалізована панель адміністратора.

Курсовий проект було створено з використанням мов програмування PHP, JavaScript, мови розмітки HTML, мови стилів CSS, систем керування базами даних MySQL. Розробка проводилась в середовищах PHPStorm, та сервером Apache.

Отже, під час виконання курсової роботи було покращено та отримано нові знання і навички з WEB – програмування: як клієнтського так і серверного. Здобуто досвід роботи зі скриптовими мовами програмування, побудови реальних моделей баз даних та роботи з ними.

Список використаних джерел

1. Інформаційні системи і технології в економіці: Навчальний посібник/ В.С. Пономаренко Р.К., Бутова І.В. Журавльова, Г.Н. Назарова та ін.; за ред. В.С. Пономаренка. - Київ: Академія, 2002. - 542с.
2. Книга про HTML <http://htmlbook.ru/>
3. Курс лекцій по HTML для початківців**.**<http://distancelearning.nuft.edu.ua/freecourses/html/index.html>
4. Лаура Томсон, Люк Веллинг Разработка веб-приложения на PHP и MySQL. - ДиаСофтЮП, 2003. – 872 с.
5. Велихов  С. HTML для тех, кто в танке**.**[http://mediaray-blog.ru](http://mediaray-blog.ru/)
6. Мержевич В. Справочник по CSS.[http://mediaray-blog.ru](http://mediaray-blog.ru/)
7. Мазуркевич.А. PHP настольная книга программиста [http://mediaray-blog.ru](http://mediaray-blog.ru/)
8. Храмцов П. Введение в программирование на JavaScript**.**[http://mediaray-blog.ru](http://mediaray-blog.ru/)
9. Вільна енциклопедія Вікіпедія  <http://uk.wikipedia.org/>
10. Веб-студія - <http://www.webostudio.com/ua/stats/CSS/>
11. Сайт Apache <http://httpd.apache.org/>
12. Сайт PHP <http://www.php.net/>
13. Сайт про  javascript    <http://javascript.ru/>

Додаток

Лістинг програмного модуля

**НТУУ «КПІ ім. І.Сікорського» ТЕФ АПЕПС ТВ-61**

**Київ – 2018**