

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка інформаційної системи
«Помічник бармена»

Виконав студент 2 курсу групи К-25
Спеціальність 122 комп'ютерні науки
Жуган Петро Миколайович

Керівник асистент
Бучинська Ірина Вікторівна

Консультант к.геогр.н., доцент
Коваленко Людмила Борисівна

Рецензент к.т.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2020

ЗМІСТ

Скорочення та умовні позначки.....	6
Вступ.....	7
1 Опис предметної області	8
Характеристика предметної області	8
Аналіз веб-сайтів	8
Характеристика сайту-візитки	9
Корпоративний сайт	10
Огляд промо-сайтів	11
Огляд сайту-вітрини.....	12
Опис інтернет-магазину.....	13
Статичні та динамічні Web-сайти	14
2 Основи роботи с базами даними.....	15
Призначення та функції систем керування базами даних	20
Основні об'єкти бази даних.....	23
Пошук інформації у базі даних	25
Аналіз основних типів запитів.....	27
3 Обґрунтування вибору мов та засобів розробки.....	29
Вимоги до Web-сайту.....	29
Мова розмітки для створення веб-сторінок	30
Основні функціональні поняття HTML.....	32
Особливості CSS та його застосування	34
Препроцесор гіпертексту	36
Опис середовища розробки	38
Аналіз програмного середовища для створення серверу	39
Аналіз середовища для керування базами даних	40
4 Розробка дизайну WEB-сайту.....	41
Опис стилю мінімалізму	41
Фізична структура сайту	47

	5
Висновки	49
Перелік джерел та посилання.....	50
Додаток Код програми	51

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

DDL – Data Definition Language;

DML – Data Manipulation Language;

HTML – Hypertext Markup Language;

DTD – Document type definition;

XHTML – EXtensible HyperText Markup Language;

UTF-8 – Unicode transformation format;

PHP – Hypertext Preprocessor;

CSS – Cascading Style Sheets;

SQL – Structured Query Language;

WAL – Write Ahead Log;

WWW – World Wide Web.

ВСТУП

В сучасному світі актуальним є веб-сайти, які можуть розміщувати певну інформацію. Веб-сайти розміщуються у мережі інтернет та майже постійно доступні, що надає можливість отримувати інформацію користувачем у кожному кутку миру улюбий час. У даному випадку для наповнення контенту веб-сайту було обрано тематику громадського харчування, що теж відповідає актуальності у сучасному світі.

Сайти, присвячені організаціям громадського харчування, безумовно, мають схожі риси. Однак у закладів різного типу є і свої особливості, які важливо враховувати при розробці веб-ресурсу.

У бари приходять люди, які прагнуть добре провести час і відпочити після важкого трудового дня. Дуже важливо, щоб на сайті бару була вказана вся інформація про те в яких чином в закладах можна розважитися або навчитися робити коктейлі. У структурі сайту має бути розміщено меню з напоями, а також сторінка з рецептами коктейлів.

Метою кваліфікаційної роботи є розробка віртуальної барної академії, де відвідувачі постійно вдосконалюються, отримуючи нові знання й беручи участь у цікавих проектах.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- розробка дизайну;
- верстка сторінками;
- веб-програмування;
- наповнення контентом;
- наявність бази даних;
- наявність карти.

Кваліфікаційна робота містить Вступ, 4 глави, висновки, 18 рисунків, 10 посилань та висновки.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Характеристика предметної області

Предметна область – це сфера застосування конкретної бази даних.

Перш ніж приступити до реалізації поставленого завдання, потрібно проаналізувати предметну область даної роботи.

Веб-сайт є візитною карткою бару, ресторану або кафе. Логічно, що він повинен виставляти заклад в правильному і хорошому світлі – бути простим, зручним. Кожна деталь веб-сайту проводить аналогію з самим баром.

Аналіз веб-сайтів

Складно переоцінити значення сайтів в своєчасному світі, адже саме завдяки їм люди отримали чудові умови для навчання, роботи і, звичайно ж, розваг. Саме на сайтах можна знайти неосяжне кількість корисної та цікавої інформації, весело провести час на розважальних сайтах, поспілкуватися з друзями в соціальних мережах, знайти хороший фільм, серіал або музичні композиції. Сайти з'явилися порівняно недавно, а саме в 1990 році, що за мірками історії взагалі дрібниця. У той час сайти вважалися дивиною, і представляли вони простенькі сторінки, часом розбавлені гіперпосиланнями для переходу на інші сторінки.

У теперішній час сайти стали невід'ємною частиною переважної більшості людей, адже саме в мережі ми проводимо чималу частину свого життя. Назва веб-сайт (website) складається з двох слів web, що означає павутина або мережу, і site – місце. Тобто фактично веб-сайт це місце у всесвітній павутині, де зібрана певна інформація. Вся інформація, яка знаходиться на сайті, розміщена на потужних серверах, що забезпечують швидкий і безперебійний доступ до сайту і всієї його інформації та контенту. Ці сервера називаються хостингом, тому в мережі досить часто можна зустріти словосполучення хостинг сайтів або хостинг купити.

Працездатність сайтів залежить не тільки від прямолінійності рук програміста і web-дизайнера, але і від надійності самого хостингу. Існує всього 2 види хостингу – платний і безкоштовний, і в залежності від цього їх характеристики можуть істотно відрізнятись. Для управління контентом сайтів можуть використовуватися різні системи управління, які вибираються в залежності від виду сайту. Існує безліч видів сайтів, починаючи від самих простеньких односторінкових сайтів і закінчуючи величезними порталами, які щодня відвідують сотні тисяч користувачів по всьому світу.

На сьогоднішній час власний інтернет-ресурс – одна із основних умов ведення успішного бізнесу. [1]¹⁾

Офіційної класифікації веб-сайтів немає, але є відмінні характеристики.

Характеристика сайту-візитки

Саме той набір інформації, поширення якої в Інтернеті, безсумнівно, корисно як новачкам, так і акулам бізнесу. Це своєрідний довідник про фірму, який містить всі необхідні контактні дані та інформацію про діяльність компанії. Це повинна бути коротка, чітка і найважливіша інформація сфери діяльності, інтереси. Не варто перевантажувати сайт-візитку інформацією, яка можливо не знадобиться потенційному клієнтові або партнерові.

Сайт-візитка – це:

- зручний засіб для сканування ґрунту ринку;
- не дорого;
- швидко;
- вигідний для організацій, у яких бізнес здійснюється не в Інтернеті.

¹⁾ [1] Глівенко С.В., Лапін Є.В., Павленко О.О. Інформаційні системи в менеджменті, навчальний посібник. ВТД «Університетська книга», Суми. 2003. 352 с.

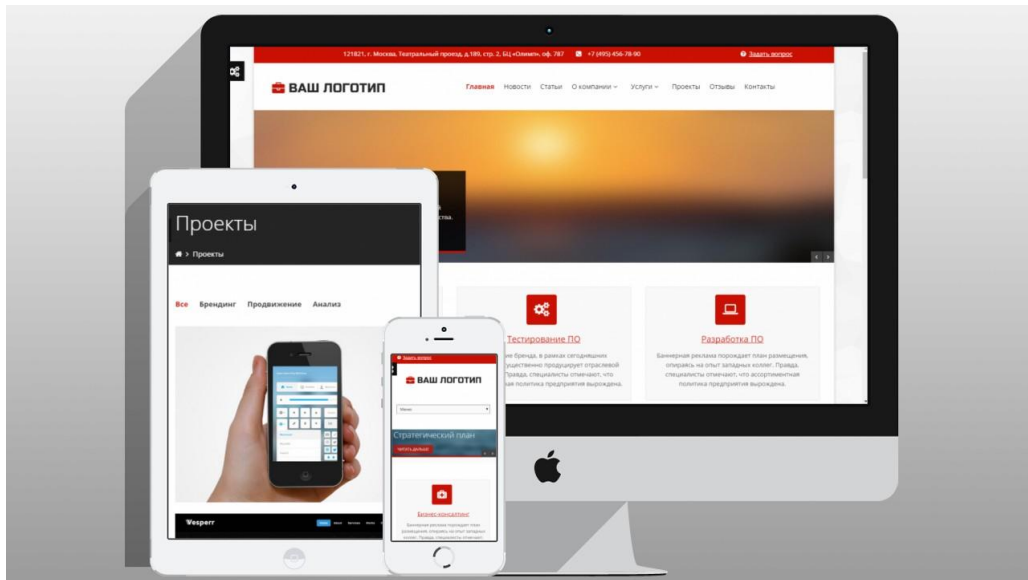


Рисунок 1 – Сайт-візитка.

Корпоративний сайт

Більш серйозним виглядом представництва компанії в Інтернеті є корпоративні сайти. Це обов'язковий інструмент успішного бізнесу будь-якої компанії. У першу чергу, сайти подібного роду допомагають формувати позитивний імідж компанії, залучають нових замовників, які можуть не знати про існування фірми на ринку, і відповідно, розширюють ринок збуту компанії. За допомогою корпоративного сайту клієнти та замовники отримують інформацію про ціни на товари і послуги в зручному вигляді. Значно збільшити прибуток є шлях надання можливості робити замовлення клієнтам прямо в Інтернеті, на головній сторінці.

Отже, корпоративний сайт – це:

- доказ серйозності і перспективності бізнесу;
- гідне представництво компанії на ринку;
- найдешевший і наймасовіший вид реклами бізнесу;
- зручно і комерційно виправдано.

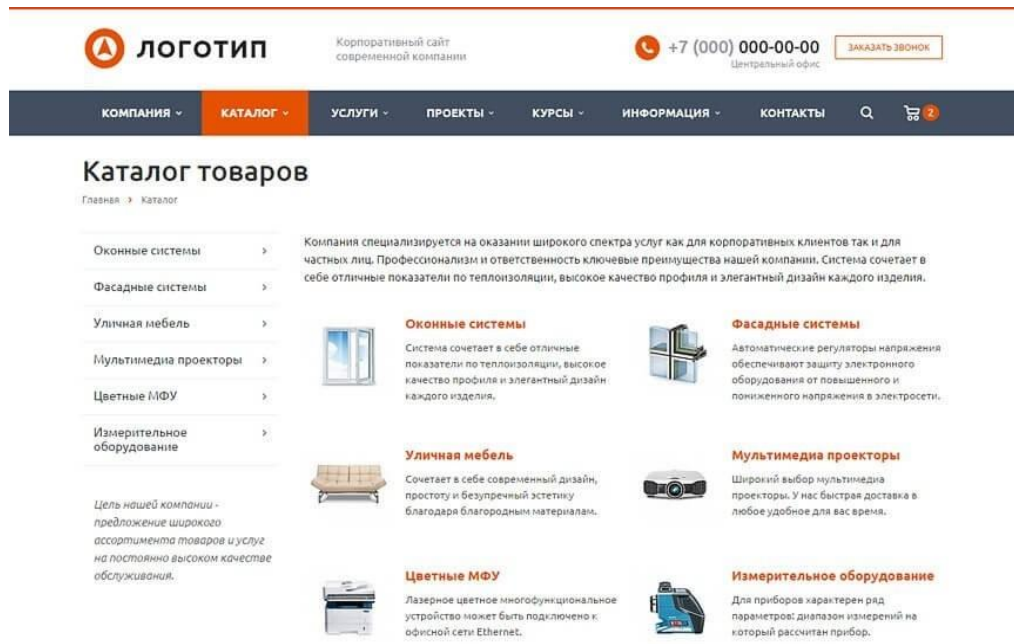


Рисунок 2 – Корпоративний сайт

Огляд промо-сайтів

Прекрасним рекламним інструментом може служити промо-сайт. Особливо, якщо власник планує провести рекламну компанію окремого товару або послуги. Структура промо-сайту сама по собі повинна виконувати функцію рекламного проспекту або буклету, і привертати увагу клієнтів. Також на такому сайті можна розмістити ту інформацію, яка не поміститься на сторінці журналу, в радіоповідомленні, телевізійному ролику. Обсяг промо-сайту – до 10 сторінок, на яких можна розмістити дані про компанію, напрямки діяльності, просування послуги, контактні дані. Окремо повідомляється інформація про акцію, із зазначенням термінів і місця проведення, умов участі, новин та інші.

Промо-сайт – це:

- використання можливостей Інтернету для впливу на цільову аудиторію;
- донесення до потенційних клієнтів максимум інформації про товар і послугу;

– потужний й ефективний інструмент реклами компанії та її продуктів.



Рисунок 3 – Промо-сайт

Огляд сайту-вітрини

Бюджетним варіантом реклами компанії є сайт-вітрина. Якщо компанія хоче продавати в Інтернеті один конкретний товар або групу товарів, то цей тип сайтів ідеально підходить для відвідувача. На таких сторінках знаходиться вся інформація, необхідна для того, щоб клієнт побачив, зацікавився і купив даний товар, не відволікаючись на новини та посилання. Це структурований каталог продукції компанії. Але цей каталог організований таким чином (фото, опис, ціна), щоб клієнт зробив як можна менше взаємодій, для того, щоб знайти потрібний товар. Головне призначення сайту-вітрини - продавати.

Сайт-вітрина – це:

- максимальний прибуток при мінімальному вкладенні сил та засобів;
- стабільний дохід;
- позитивний імідж компанії.



Рисунок – 5 Сайт-вітрина

Опис інтернет-магазину

На відміну від сайту-вітрини, де клієнт може побачити наявність всіх товарів, за допомогою інтернет-магазину клієнт може ще й зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату. Такий сайт має бути зручним і функціональним, щоб відвідувач міг легко знайти те, що йому потрібно, відправити товар у кошик та оформити покупку в кілька взаємодій. Інтернет-магазин істотно зменшить витрати компанії, так, як не треба орендувати приміщення для магазину, не треба платити заробітну плату співробітникам. Серед інших переваг: величезна аудиторія, можливість розміщувати необмежений асортимент товарів, своєчасно реагувати на зміни на ринку, враховувати потреби клієнтів та інші.

Інтернет-магазин – це:

- нові можливості для бізнесу;
- розширення ринку збуту;
- залучення широкої аудиторії клієнтів;
- скорочення витрат.

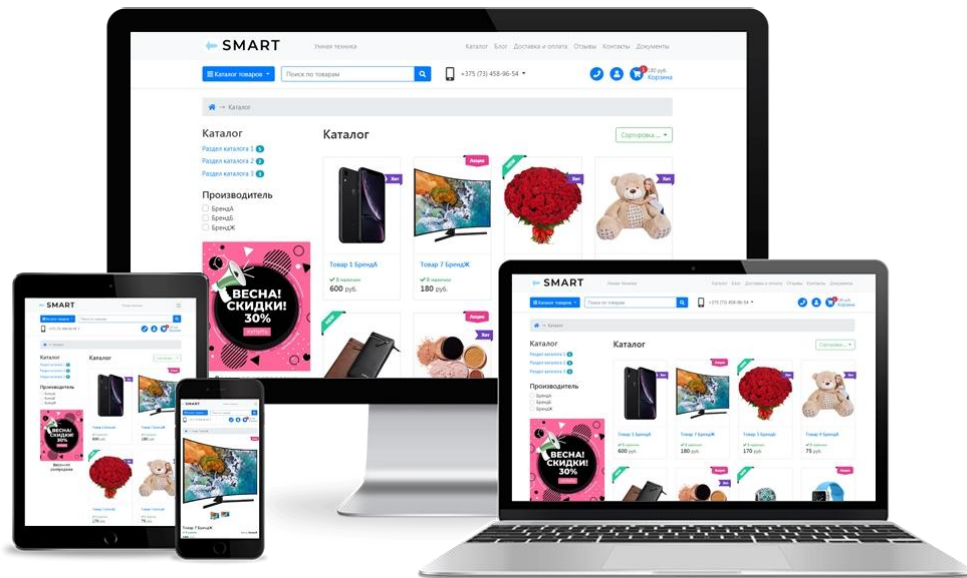


Рисунок 6 – Інтернет-магазин

Статичні та динамічні Web-сайти

Крім класифікації сайтів за характеристиками можна розрізнити веб-сторінки і за складністю. Всього існує два види: статичний або динамічний сайт.

Так, статичним називають сайт з набором постійних, незмінних сторінок. Наприклад, сайт-візитка або сайт-вітрина. Інформація на таких сторінках жорстко закріплена, і зміни можуть здійснюватися тільки фахівцем у веб-технологіях. Такі сайти швидко завантажуються, можуть переглядатися в браузері, мають мінімальні вимоги до веб-сервера. Для компаній такі сайти – бюджетне рішення. Але для підтримки та актуалізації інформації на сторінці все одно будуть потрібні фахівці.

У цьому плані, динамічний сайт має безліч переваг. Для функціонування подібних веб-сторінок використовується система CMS (система управління змістом). Це забезпечення, яке дозволяє оновлювати інформацію, публікувати і змінювати її без залучення фахівців. Досить провести навчання персоналу лише в області порядку роботи з системою.

2 ОСНОВИ РОБОТИ С БАЗАМИ ДАНИМИ

У наші дні люди часто говорять про бази даних [2]¹⁾. Комп'ютери становлять невід'ємну частину сучасного суспільства, тому нерідко можна почути фрази начебто «Я пошукаю твій запис у базі даних». І мова йде не про великі ящики, де зберігаються купи каталогів, а про комп'ютерні системи, призначені для прискореного пошуку інформації.

Комп'ютери так тісно увійшли у наше життя, що їх можна запрограмувати на виконання рутинних, повторюваних операцій і вирішення завдань, які нам самим було б не під силу вирішити без їхньої обчислювальної швидкості і ємності інформаційних носіїв. Розміщення інформації на папері і розроблення схеми зберігання паперів у каталогах і картотеках – досить чітко відпрацьований процес, але багато хто зітхнув з полегшенням, коли завдання звелося до переміщення електронних документів у каталогах на жорсткому диску.

Одними з функцій баз даних є впорядкування й індексація інформації. Як і у бібліотечній картотеці, не потрібно переглядати половину архіву, щоб знайти потрібний запис. Усе виконується набагато швидше.

Не всі бази даних створюються на основі тих самих принципів, але традиційно в них застосовується ідея організації даних у вигляді записів. Кожен запис має фіксований набір полів. Записи містяться у таблиці, а сукупність таблиць формує базу даних.

Для роботи з базою даних необхідна СКБД (система керування базами даних) – сукупність програмних та лінгвістичних засобів загального чи спеціального призначення, що забезпечують управління створенням та використанням бази даних.

СКБД управляє однією або декількома базами даних. База даних являє собою сукупність інформації, організованої у вигляді множин. Кожна множина містить записи уніфікованого виду. Самі записи складаються з полів. Як правило, множини називають таблицями, а записи - рядками таблиць. Така

¹⁾ [2] Макарова М.В. Інформатика та комп'ютерна техніка, навчальний посібник. ВТД «Університетська книга», Суми. 2003. 642 с.

логічна модель даних. На жорсткому диску вся база даних може перебувати в одному файлі.

Рядки таблиць можуть бути зв'язані один з одним одним із трьох способів. Найпростіше співвідношення – «один до одного». У цьому випадку рядок першої таблиці відповідає одному єдиному рядку другої таблиці. На діаграмах таке співвідношення виражається записом 1:1.

Співвідношення «один до багатьох» означає ситуацію, коли рядок однієї таблиці відповідає декільком рядкам іншої таблиці. Це найпоширеніший тип співвідношень. На діаграмах він виражається записом 1:N.

Нарешті, при співвідношенні «багато хто до багатьох» рядки першої таблиці можуть бути пов'язані з довільним числом рядків у другій таблиці. Таке співвідношення записується як N:M.

Ієрархічні бази даних підтримують деревоподібну організацію інформації. Зв'язки між записами виражаються у вигляді відносин предок/нащадок, а в кожного запису є тільки один батьківський запис. Це допомагає підтримувати цілісність посилань. Коли запис видаляється з дерева, всі його нащадки також повинні бути вилучені.

Ієрархічні бази даних мають централізовану структуру, тобто безпеку даних легко контролювати. На жаль, певні знання про фізичний порядок зберігання записів все-таки необхідні, тому що відносини предок/нащадок реалізуються у вигляді фізичних покажчиків з одного запису на інші.

Це означає, що пошук запису здійснюється методом прямого обходу дерева. Записи, розміщені в одній половині дерева, шукаються швидше, ніж в іншій. Звідси виникає необхідність правильно впорядковувати записи, щоб час їх пошуку був мінімальним. Це важко, тому що не всі відношення, які існують у реальному світі, можна виразити в ієрархічній базі даних. Відношення «один до багатьох» є природними, але практично неможливо описати відношення «багато хто до багатьох» або ситуації, коли запис має кілька предків. Доти, поки у додатках будуть кодуватися відомості про фізичну структуру даних, будь-які зміни цієї структури будуть призводити до перекомпіляції.

Мережні бази даних. Мережна модель розширює ієрархічну модель, дозволяючи групувати зв'язки між записами у множини. З логічної точки зору зв'язок - це не сам запис. Зв'язки лише виражають відношення між записами. Як і в ієрархічній моделі, зв'язки ведуть від батьківського запису до дочірнього, але цього разу підтримується множинне успадковування.

Відповідно до специфікації CODASYL мережна модель підтримує DDL (Data Definition Language – мова визначення даних) і DML (Data Manipulation Language – мова обробки даних). Це спеціальні мови, призначені для визначення структури бази даних і складання запитів. Незважаючи на їх наявність, програміст, як і раніше, повинен знати структуру бази даних.

У мережній моделі допускаються відношення «багато хто до багатьох», а записи не залежать один від одного. При видаленні запису віддаляються й всі його зв'язки, але не самі зв'язані записи.

У мережній моделі потрібно, щоб зв'язки встановлювалися між створеними записами, для того щоб уникнути дублювання й перекручування цілісності. Дані можна ізолювати у відповідних таблицях і зв'язати із записами в інших таблицях.

Програмістові не потрібно піклуватися про те, як організується фізичне зберігання даних на диску. Це послабляє залежність додатків і даних. Але в мережній моделі потрібно, щоб програміст пам'ятав структуру даних при формуванні запитів.

Оптимальну структуру бази даних складно сформувати, а готову структуру важко змінювати. Якщо вид таблиці зазнає змін, усі відношення з іншими таблицями повинні бути встановлені заново, щоб не порушилася цілісність даних. Складність подібного завдання приводить до того, що програмісти найчастіше скасовують деякі обмеження цілісності заради спрощення додатків.

Реляційні бази даних. У реляційній моделі база даних являє собою централізоване сховище таблиць, що забезпечує безпечний одночасний доступ до інформації з боку багатьох користувачів. У рядках таблиць частина

полів містить дані, що стосуються безпосередньо запису, а частина - посилання на записи інших таблиць. Таким чином, зв'язки між записами є невід'ємною частиною реляційної моделі.

Кожен запис таблиці має однакову структуру. Наприклад, у таблиці, що містить опис автомобілів, у всіх записів буде той самий набір полів: виробник, модель, рік випуску, пробіг і так далі. Такі таблиці легко зображувати у графічному вигляді.

У реляційній моделі досягається інформаційна й структурна незалежність. Записи не зв'язані між собою настільки, щоб зміна одного з них стосувалася інших, а зміна структури бази даних не обов'язково приводить до перекомпіляції сумісних з нею додатків.

У реляційних СКБД застосовується мова SQL, що дозволяє формулювати довільні, нерегламентовані запити. Це мова четвертого покоління, тому будь-який користувач може швидко навчитися робити запити. До того ж, існує безліч додатків, що дозволяють будувати логічні схеми запитів у графічному вигляді. Все це відбувається за рахунок жорсткості вимог до продуктивності комп'ютерів. На щастя, сучасні обчислювальні потужності більш ніж адекватні.

У реляційних базах даних можуть бути розходження у реалізації мови SQL, хоча це й не проблема реляційної моделі. Кожна реляційна СКБД реалізує якусь підмножину стандарту SQL плюс набір унікальних команд, що ускладнює завдання програмістам, які намагаються перейти від однієї СКБД до іншої. Доводиться робити нелегкий вибір між максимальною сумісністю й максимальною продуктивністю. У першому випадку потрібно дотримуватися мінімального загального набору команд, підтримуваних у кожній СКБД. У другому випадку програміст просто зосереджується на роботі в даній конкретній СКБД, використовуючи переваги її унікальних команд і функцій.

Об'єктно-орієнтована база даних (ООБД) дозволяє програмістам, які працюють із мовами третього покоління, інтерпретувати всі свої інформаційні сутності як об'єкти, що зберігаються в оперативній пам'яті. Додатковий

інтерфейсний рівень абстракції забезпечує перехоплення запитів, що стосуються тих частин бази даних, які перебувають у постійному сховищі на диску. Зміни, внесені в об'єкти, оптимальним чином переносяться з пам'яті на диск.

Перевагою об'єктно-орієнтованих баз даних є спрощений код. Додатки одержують можливість інтерпретувати дані в контексті тієї мови програмування, на якій вони написані. Реляційна база даних повертає значення всіх полів у текстовому вигляді, а потім вони зводяться до локальних типів даних. В об'єктно-орієнтованих базах даних цей етап ліквідований. Методи маніпулювання даними завжди залишаються однаковими незалежно від того, перебувають дані на диску або в пам'яті.

Дані в об'єктно-орієнтованих базах даних здатні на вигляд будь-якої структури, яку можна виразити використовуваною мовою програмування. Відношення між сутностями також можуть бути досить складними. Об'єктно-орієнтована база даних управляє кеш-буфером об'єктів, переміщуючи об'єкти між буфером і дисковим сховищем у міру необхідності.

За допомогою об'єктно-орієнтованих баз даних вирішуються дві проблеми. По-перше, складні інформаційні структури виражаються в них краще, ніж у реляційних базах даних, а по-друге, усувається необхідність транслювати дані з того формату, що підтримується в СКБД. Наприклад, у реляційній СКБД розмірність цілих чисел може становити 11 цифр, а у використовуваній мові програмування – 16. Програмістові потрібно враховувати цю ситуацію.

Об'єктно-орієнтовані СКБД виконують багато додаткових функцій. Це окупається сповна, якщо відношення між даними дуже складні. У такому разі продуктивність об'єктно-орієнтованих баз даних виявляється вищою, ніж у реляційних СКБД. Якщо ж дані менш складні, додаткові функції виявляються надлишковими. В об'єктній моделі даних підтримуються нерегламентовані запити, але мовою їх складання необов'язково є SQL. Логічне подання даних може не відповідати реляційній моделі, тому застосування мови SQL стане безглуздом. Найчастіше зручніше обробляти об'єкти в пам'яті, виконуючи відповідні види пошуку.

Великим недоліком об'єктно-орієнтованих баз даних є їх тісні зв'язки із застосовуваною мовою програмування. До даних, що зберігаються в реляційній СКБД, можуть належати будь-які додатки, тоді як, приміром, Java-об'єкт, поміщений в об'єктно-орієнтовану базу даних, буде становити інтерес лише для додатків, написаних на Java.

Об'єктно-реляційні бази даних. Об'єктно-реляційні СКБД поєднують у собі риси реляційної й об'єктної моделей. Їх виникнення породжене тим, що реляційні бази даних добре працюють із убудованими типами даних і набагато гірше - з користувацькими, нестандартними. Коли з'являється новий важливий тип даних, доводиться або включати його підтримку в СКБД, або змушувати програміста самостійно управляти даними в додатку.

Не всяку інформацію має сенс інтерпретувати у вигляді ланцюжків символів або цифр. Уявимо собі музичну базу даних. Пісню, закодовану у вигляді аудіофайлу, можна помістити в текстове поле великого розміру, але як у такому випадку буде здійснюватися текстовий пошук?

Перебудова СКБД із метою включення в неї підтримки нового типу даних не кращий вихід з положення. Замість цього об'єктно-реляційна СКБД дозволяє завантажувати код, призначений для обробки "нетипових" даних. Таким чином, база даних зберігає свою табличну структуру, але спосіб обробки деяких полів таблиць визначається ззовні, тобто програмістом.

Призначення та функції систем керування базами даних

Діяльність з пошуку прийнятних засобів для автоматизованої обробки інформації сприяла створенню на початку 60-х років 20 ст. спеціальних програмних комплексів, "Систем управління базами даних" (СКБД), які повинні надавати користувачеві способи введення, збереження даних, а також опису структури даних.

СКБД повинна також надавати механізми пошуку даних за запитом користувачів, забезпечувати захист даних від некоректного оновлення та несанкціонованого доступу, підтримувати бази даних в актуальному стані.

Інформація в реляційних базах даних зберігається у табличній формі. Це може бути одна або декілька таблиць. Під час роботи з даними декількох таблиць встановлюються зв'язки між таблицями.

Сучасні СКБД забезпечують:

- набір засобів для підтримки таблиць і співвідношень між зв'язаними таблицями;
- введення, модифікацію інформації, пошук і подання інформації у текстовому або графічному вигляді;
- засоби програмування, за допомогою яких ви можете створювати власні додатки.

Отже, для створення бази даних і роботи з нею, крім комп'ютера, потрібно мати відповідний програмний продукт – СКБД.

До функції СКБД прийнято відносити наступні параметри.

Безпосереднє управління даними у зовнішній пам'яті. Ця функція включає забезпечення необхідних структур зовнішньої пам'яті як для зберігання даних, що безпосередньо входять у БД, так і для службових цілей, наприклад, для прискорення доступу до даних в деяких випадках (звичайно для цього використовуються індекси). У деяких реалізаціях СКБД активно використовуються можливості існуючих файлових систем, в інших робота проводиться аж до рівня пристроїв зовнішньої пам'яті. Але користувачі у будь-якому випадку не зобов'язані знати, чи використовує СКБД файлову систему і якщо використовує, то як організовані файли. Зокрема, СКБД підтримує власну систему іменування об'єктів БД.

Управління буферами оперативної пам'яті. СКБД, як правило, працюють з БД значного розміру; принаймні цей розмір звичайно істотно більший за доступну ємність оперативної пам'яті. Зрозуміло, що якщо при зверненні до будь-якого елемента даних проводитиметься обмін із зовнішньою пам'яттю, то вся система працюватиме із швидкістю пристрою зовнішньої пам'яті. Практично єдиним способом реального збільшення цієї швидкості є буферизація даних в оперативній пам'яті.

Управління транзакціями. Транзакція – це послідовність операцій над БД, що розглядаються СКБД як єдине ціле. Або транзакція успішно виконується, і СКБД фіксує зміни БД, проведені цією транзакцією, в зовнішній пам'яті, або жодна з цих змін ніяк не відображається на стані БД. Поняття транзакції необхідне для підтримки логічної цілісності БД.

Журналізація. Однією з основних вимог до СКБД є надійність зберігання даних у зовнішній пам'яті. Під надійністю зберігання розуміють те, що СКБД повинна бути у змозі відновити останній узгоджений стан БД після будь-якого апаратного або програмного збою. Звичайно розглядаються два можливі види апаратних збоїв: так звані м'які збої, які можна трактувати як раптовий зупин роботи комп'ютера (наприклад, аварійне виключення живлення), і жорсткі збої, що характеризуються втратою інформації на носіях зовнішньої пам'яті.

У будь-якому випадку для відновлення БД потрібно мати у своєму розпорядженні деяку додаткову інформацію. Іншими словами, підтримка надійності зберігання даних у БД вимагає надмірності зберігання даних, причому та частина даних, яка використовується для відновлення, повинна зберігатися особливо надійно. Найбільш поширеним методом підтримки такої надмірної інформації є ведення журналу змін БД.

Журнал – це особлива частина БД, недоступна користувачам СКБД і підтримувана з особливою ретельністю (іноді підтримуються дві копії журналу, що розміщуються на різних фізичних дисках), до якої надходять записи про всі зміни основної частини БД. У різних СКБД зміни БД журналізуються на різних рівнях: іноді запис у журналі відповідає деякій логічній операції зміни БД (наприклад, операції видалення рядка з таблиці реляційної БД), іноді мінімальної внутрішньої операції модифікації сторінки зовнішньої пам'яті; у деяких системах одночасно використовуються обидва підходи.

У всіх випадках дотримуються стратегії "попереднього" запису в журнал (так званого протоколу Write Ahead Log – WAL). Ця стратегія полягає у тому, що запис про зміну будь-якого об'єкта БД повинен потрапити у зовні-

шню пам'ять журналу раніше, ніж змінений об'єкт потрапить у зовнішню пам'ять основної частини БД. Відомо, що якщо в СКБД коректно дотримується протокол WAL, то за допомогою журналу можна вирішити усі проблеми відновлення БД після будь-якого збою.

Найпростіша ситуація відновлення - індивідуальний відкат транзакції. Строго кажучи, для цього не потрібен загальносистемний журнал змін БД.

Для відновлення БД після жорсткого збою використовують журнал і архівну копію БД.

Архівна копія – це повна копія БД до моменту початку заповнення журналу.

Підтримка мов БД. Для роботи з базами даних використовуються спеціальні мови, у цілому звані мовами баз даних.

У сучасних СКБД, як правило, підтримується єдина інтегрована мова, що містить усі необхідні засоби для роботи з БД, починаючи від її створення, і забезпечує базовий призначений для користувача інтерфейс з базами даних. Стандартною мовою найбільш поширених у даний час реляційних СКБД є мова SQL (Structured Query Language).

Основні об'єкти бази даних

До об'єктів бази даних належать таблиці.

Таблиці – це набір даних з однієї визначеної таблиці теми. Наприклад, дані про студентів, їх прізвища, адреси, телефони. Рядки двовимірних таблиць називають записами, стовпчики – полями. У термінах реляційних СКБД подібні таблиці називаються відношеннями, їх записи – кортежі відношень, поля – атрибути відношень. Записи відрізняються своїм номером, а поля – своїм ім'ям.

Основні умови щодо змісту таблиць такі:

- однакові записи забороняються;
- всі записи повинні мати однакову кількість полів;

- значення полів атомарні, тобто таблиця не може мати своїми компонентами інші таблиці.

Форми використовуються для введення нових даних і перегляду існуючих у вказаному форматі. Можна використовувати кнопку форму для відкриття інших форм або звітів. Форми – це основний засіб побудови інтерфейсу користувача, що забезпечує найбільш зручний спосіб перегляду та редагування даних, а також контроль за ходом виконання прикладної програми. Таким чином, форми будуються для:

Виведення та редагування даних. Це найбільш поширений спосіб використання форм. Вони забезпечують виведення на екран монітора даних у необхідному вигляді. Використання таких форм значно спрощує редагування даних, їх введення та видалення з бази даних. При цьому деякі дані можна зробити доступними лише для перегляду, а деякі зовсім не демонструвати. Крім цього, є можливість у таких формах забезпечити процес обчислення полів в залежності від параметрів, які задає користувач;

Керування ходом виконання прикладної програми. Сучасні прикладні програми, як правило, мають оболонку для надання користувачеві можливості виконувати ті чи інші дії у певній послідовності. Таку оболонку можна побудувати за допомогою форм, а саме: кнопочових форм. Такі форми містять кнопки, що викликають при їх виборі дію певного макросу або процедури VBA. Ці кнопки називаються командними. Командні кнопки можуть використовуватись і для виклику іншої форми, а також багатьох інших дій, що пов'язані з ходом виконання прикладної програми, а саме: виконувати запити, команди меню, фільтрувати дані, друкувати звіти тощо;

Введення даних. Можна побудувати форми, які призначені лише для введення в базу даних нових значень, які автоматизують виконання програми за певним алгоритмом;

Виведення повідомлень. Такі форми забезпечують інформацією події, що пов'язані з виконанням прикладної програми, наприклад, повідомлення про різноманітні помилки;

Друк інформації. Як правило, для друку призначені звіти, але надрукувати інформацію можна і за допомогою форм. Очевидно для введення та виведення даних діють різні параметри, тому у таких формах інструментально підтримується їх подвійна роль.

Запити-вимоги на відбір даних, що зберігаються у таблицях, або вимога на виконання певних дій з даними. За допомогою запитів дані упорядковують, фільтрують, об'єднують, аналізують.

Звіти – це об'єкти бази даних Microsoft Access, призначені для відображення даних, організованих і відформатованих відповідно до вимог користувача. За допомогою звітів складаються комерційні відомості, списки телефонів або списки розсилання. Звіти використовуються для відображення інформації у друкованому вигляді.

Макроси містять одну або декілька макрокоманд, які використовуються для автоматичного виконання деяких операцій.

Модулі містять програми на VISUAL BASIC. Дозволяють розширити можливості системи, якщо написати для цього необхідні програми.

Пошук інформації у базі даних

У процесі роботи з інформацією у таблицях виникає потреба пошуку потрібного запису. Для цього використовується ряд спеціальних засобів, а саме за допомогою команди Правка\Найти, за допомогою фільтрів або запитів.

Значення для пошуку можна ввести так, як воно подане у полі, або користуючись спеціальними символами:

- * – будь-яка кількість будь-яких символів;
- ? – будь-який текстовий символ;
- # – будь-яка цифра або будь-який символ.

Пошук даних за допомогою фільтра. Фільтр використовується для того, щоб відобразити лише ті значення, які визначені заданими критеріями.

Існують чотири способи, які використовуються для вибору записів за допомогою фільтрів:

Фільтр за виділеним фрагментом, потрібно використовувати, якщо можна легко знайти і вибрати у таблиці те значення, яке повинне містити записи, що вибираються.

Звичайний фільтр використовується для вибору шуканих значень із списку без перегляду всіх записів у таблиці.

Поле «Фільтр для» використовується для введення конкретного значення або виразу, введеного як умова вибору в контекстне меню.

Розширений фільтр використовується для створення складних фільтрів, якщо необхідно задати ряд критеріїв вибору записів.

Для використання фільтра служить команда меню ЗАПИСИ/ ФІЛЬТР.

Запити та їх застосування. Для користувача основним режимом роботи з базою даних є режим отримання інформації. Причому нарівні з інформаційним пошуком у базі даних потрібно здійснювати перетворення (редагування, упорядкування, групування та елементарні обчислення) даних для видачі користувачеві різних довідок, звітів і тому подібне. Це виконується за допомогою запитів. Запити дозволяють переглядати, аналізувати і змінювати дані з декількох таблиць.

Основна подібність між запитом на вибірку і фільтрами полягає у тому, що в них проводиться вилучення підмножини записів із базової таблиці або запиту.

Фільтр, як правило, використовують під час роботи у режимі Форми або у режимі Таблиці для перегляду або зміни підмножини записів. На відміну від фільтра запит дозволяє отримати більш змістовний результат. Перш за все, це пояснюється тим, що фільтр дає інформацію для перегляду (друку), але на відміну від запиту автоматично не зберігається як окремий об'єкт бази даних. Запити, маючи таку властивість, дозволяють динамічно поновлювати інформацію у своїх таблицях, якщо у таблицях бази даних виникла зміна інформації. Крім цього, запит має і зворотню дію: якщо змінювати інформацію

цію у його таблицях, то таблиці бази даних, на основі яких побудований запит, будуть адекватно змінювати свою інформацію.

Запити використовують для виконання таких дій:

- перегляду підмножини записів без попереднього відкриття конкретної таблиці або форми;
- вибору таблиць, що містять записи, з можливістю подальшого додавання інших таблиць;
- виконання обчислень над значеннями полів.

Аналіз основних типів запитів

Запит на вибирання, забезпечує вибір даних із зв'язаних таблиць і таблиць, побудованих під час реалізації інших запитів. (наприклад, запит на вибирання записів про робітників, які мають вищу освіту, запит на вибирання записів про робітників, які працюють менше трьох років, і так далі). Запити на вибирання створюють тимчасові таблиці з вибіркою даних. Базові таблиці при цьому не змінюються. Запити на вибирання можна також використовувати для групування записів і обчислення сум, середніх значень, підрахунку кількості записів і отримання інших типів підсумкових значень.

Запит з параметрами відображає одне або декілька визначених діалогових вікон, які виводять запрошення користувачу ввести умови вибирання. При запуску такого запиту можна ввести критерії вибирання записів або значення для вставки у поле.

Перехресний запит відображає результати статистичних розрахунків (такі, як суми, кількість записів і середні значення), виконаних за даними з одного поля таблиці. Ці результати групуються у вигляді таблиці по двох наборах даних, один з яких визначає заголовки стовпчиків, а інший заголовки рядків. Перехресні запити використовують для розрахунків і подання даних у структурі, яка полегшує їх аналіз.

Запит на внесення змін за одну операцію вносить зміни або переміщує декілька записів. Існує чотири типи таких запитів.

Запит на створення таблиці ґрунтується на запиті на вибирання і забезпечує формування і створення нової таблиці на основі усіх або частини даних із однієї чи декількох таблиць.

Запит на оновлення дає змогу вносити зміни у групу записів, які вибираються за допомогою запиту на вибирання. (підвищити заробітну плату деякій категорії робітників, наприклад, водіям або робітникам, які мають ставку, меншу від зазначеної суми).

Запит на знищення забезпечує виключення записів з однієї або кількох зв'язаних таблиць (наприклад, ліквідувати на підприємстві деяку посаду).

Запит на додавання записів. Даний запит додає групу записів із однієї або декількох таблиць у кінець однієї або декількох таблиць.

Для однієї і тієї самої таблиці можна створювати різні запити, кожен з яких може вилучати з таблиці певну частину інформації, яка в даний момент необхідна.

3 ОБҐРУНТУВАННЯ ВИБОРУ МОВ ТА ЗАСОБІВ РОЗРОБКИ

Вимоги до Web-сайту

Сайт повинен відповідати таким вимогам, як:

- мати загальну інформацію про коктейлі та їх рецепти;
- мати інформацію про якісні барні заклади;
- мати систему реєстрації.

Вимоги до дизайну сайту:

Веб-сайт є візитною карткою бару, ресторану або кафе. Дизайн сайту кафе чи бару повинен повністю відповідати тематиці і духу закладу, бути унікальним і викликати добрі враження.

При розробці сайту дуже важливо продумати його структуру і навігацію. Структура повинна бути простою і логічною, щоб користувачі на рівні інтуїції було зрозуміло як знайти потрібну йому інформацію, як переглянути меню, подивитися гарні майстер-класи та рецепти коктейлів.

Вимоги до функціоналу сайту: при розробці сайту дуже важливою частиною являє інтеграція соціальних мереж; меню слід виносити окремою категорією. Кожен існуючий коктейль на сторінці можна прив'язати з фотографією.

Основні вимоги до дизайну сайту:

- дизайн сайту повинен бути сучасним і зручним для відвідувачів;
- на сайті не повинно бути багато зайвої інформації;
- сайт повинен мати зручне навігаційне меню;
- бути оптимізованим, мати високу швидкість завантаження і адаптивну версію;
- сайт повинен використовувати актуальні технології;
- сайт має використовувати технології захисту даних, якщо він їх оброблює.

Навігаційне меню є відображення структури сайту, яке бачить відвідувач, потрапляючи на ресурс.

Відвідувач повинен завжди мати перед очима “провідну нитку”, якою є кнопка повернення в попередній розділ (сторінку). Також на кожній сторінці сайту повинна бути розміщене посилення “На головну”, якщо відвідувач знаходиться на іншій сторінці. Як зображено на рис. 7.



Рисунок 7 – Меню

Мова розмітки для створення веб-сторінок

HTML [3]¹⁾ (Hypertext Markup Language – мова гіпертекстової розмітки) – це мова опису структури сторінок документів, яка дозволяє звичайний текст формувати в абзаци, заголовки, списки та інші структури, створювати посилення на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну інформацію. Теги повідомляють браузерам, як формувати і представляти інформацію на екрані.

Мова гіпертекстової розмітки HTML була запропонована Тімом Бернерсом – Лі у 1989 як один з компонентів технології розробки розподіленої гіпертекстової системи World Wide Web. Ідея гіпертекстової інформаційної системи полягає у тому, що користувач має можливість переглядати документи (сторінки тексту) у найбільш зручному для себе порядку, а не послідовно, як це прийнято при читанні книг. Досягається це шляхом створення

¹⁾ [3] Лабберс П. HTML5 для напівпрофесіоналів. Потужні інструменти для розробки сучасних веб-застосунків. М.: Вільямс, 2011. 272 с.

спеціального механізму пов'язування різних сторінок тексту за допомогою гіпертекстових посилань.

Мова HTML дозволяє визначити структуру електронного документа з поліграфічним рівнем оформлення. Створений документ може містити різноманітні елементи: ілюстрації, аудіо і відео фрагменти. Мова HTML включає розвинені засоби для визначення кількох рівнів заголовків, шрифтових виділень, різних груп об'єктів та багато інших можливостей.

Важливим чинником, який вплинув на розвиток мови HTML, став її вибір за основу для гіпертекстової бази даних звичайного текстового файлу, який можна створювати у будь-якому текстовому редакторі на будь-якій апаратній платформі у середовищі будь-якої операційної системи.

Таким чином, гіпертекстова база даних у концепції WWW – це набір текстових файлів, розмічених мовою HTML, яка визначає форму представлення інформації (розмітка) і структуру зв'язків цих файлів (гіпертекстові посилання).

За основу моделі розмітки документів у HTML прийнята тегова модель. Тегова модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами. Тобто документ HTML є не чим іншим, як звичайним ASCII-файлом з доданими до нього HTML-кодами (тегами).

Теги HTML-документів в основному є простими і зрозумілими для використання, оскільки вони створені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень.

Важливо підкреслити, що дану технологію використовують не тільки для ескізування сайтів. Також дану мову використовують для інтеграцій у мобільній платформі для автоматичного оброблювання гіперпосилань, атрибутів тексту та інше.

Основні функціональні поняття HTML

Як показано на прикладі HTML-коду, який міститься нижче на рисунку 8, він завжди починається з `<!DOCTYPE html>`. Для браузера поява цього елемента вказує на те, що йому слід відображати HTML-структуру:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01//EN
"http://www.w3.org/TR/html4/strict.dtd"
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>Приклад-веб сторінки</title>
</head>
<body>
<h1>Заголовок</h1>
<!-- Коментар -->
<p>Перший абзац.</p>
<p>Другий абзац.</p>
</body>
</html>
```

Елемент `<!DOCTYPE>` призначений для вказівки типу поточного документа – DTD (document type definition, опис типу документа). Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну веб-сторінку, адже HTML існує в декількох версіях, крім того, є XHTML (EXtensible HyperText Markup Language, розширена мова розмітки гіпертексту), схожий на HTML, але розрізняються з ним по синтаксису. Щоб браузер «не плутався» і розумів, згідно яким стандартом відображати веб-сторінку і необхідно в першому рядку коду задавати `<!DOCTYPE>`.

```
<html>
```

Тег `<html>` визначає початок HTML-файлу, всередині нього зберігається заголовок (`<head>`) і тіло документа (`<body>`).

```
<head>
```

Заголовок документа, як ще називають блок `<head>`, може містити текст і теги, але вміст цього розділу не показується безпосередньо на сторінці, за винятком контейнера `<title>`.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Тег `<meta>` є універсальним і додає цілий клас можливостей, зокрема, за допомогою метатегів, як узагальнено називають цей тег, можна змінювати кодування сторінки, додавати ключові слова, опис документа і багато іншого. Щоб браузер розумів, що має справу з кодуванням UTF-8 (Unicode transformation format, формат перетворення Юнікод) і додається даний рядок.

```
<title>Приклад веб-сторінки </title>
```

Тег `<title>` визначає заголовок веб-сторінки, це один з важливих елементів призначений для вирішення безліч завдань. В операційній системі Windows текст заголовка відображається в лівому верхньому кутку вікна браузера.

Тег `<title>` є обов'язковим і повинен неодмінно бути присутнім в коді документа.

```
</head>
```

Обов'язково слід додавати закриває тег `</head>`, щоб показати, що блок заголовка документа завершено.

```
<body>
```

Тіло документа `<body>` призначений для розміщення тегів і змістовної частини веб-сторінки.

```
<h1>Заголовок</h1>
```

HTML [4]¹⁾ пропонує шість текстових заголовків різного рівня, які показують відносну важливість секції, розташованої після заголовка. Так, тег `<h1>` представляє собою найбільш важливий заголовок першого рівня, а тег `<h6>` служить для позначення заголовка шостого рівня і є найменш значним. За замовчуванням, заголовок першого рівня відображається найбільшим шрифтом жирного накреслення, заголовки подальшого рівня за розміром менше. Теги `<h1>` ... `<h6>` відносяться до блокових елементів, вони завжди починаються з нового рядка, а після них інші елементи відображаються на наступному рядку. Крім того, перед заголовком і після нього додається порожній простір.

¹⁾ [4] Джамса К. Ефективний самовчитель по креативному Web-дизайну. СПб: ДиаСофтЮп, 2005. 672 с.

`<p>Перший абзац.</p>`

Тег `<p>` визначає абзац (параграф) тексту. Якщо тег не було закрито, вважається, що кінець абзацу збігається з початком наступного блокового елемента.

`<p>Другий абзац.</p>`

Тег `<p>` є блоковим елементом, тому текст завжди починається з нового рядка, абзаци йдуть один за одним розділяються між собою відбиттям (так називається порожній простір між ними).

`</body>`

Слід додати, що тег `</body>`, необхідний для того, щоб вказати, що тіло документа завершено.

Останнім елементом в коді завжди йде тег `</html>`, який службовою міткою закінчення гіпертекстового документу.

Особливості CSS та його застосування

Основним поняттям CSS [5]¹⁾ (Cascading Style Sheets) є стиль – набір правил оформлення і форматування, який може бути застосований до різних елементів сторінки. У стандартному HTML для присвоєння елементу певних властивостей (таких, як колір, розмір, розташування на сторінці і т. д.) доводилося кожного разу описувати ці властивості, навіть якщо на одній сторінці повинно розташовуватися десятки однакових елементів.

CSS діє іншим, зручнішим і економнішим способом. Для надання певних характеристик необхідному елементу потрібно один раз описати цей елемент і визначити цей опис як стиль, а надалі просто вказувати, що елемент, який необхідно оформити відповідним чином, повинен прийняти властивості описаного стилю. Застосувати таблицю стилів до HTML-документу можна трьома способами: застосувати зовнішні стилі (у вигляді окремого текстового. CSS-файлу) за допомогою елемента `link` вбудувати стилі безпосередньо HTML документ(у вигляді блоку `css` тексту) за допомогою

¹⁾ [5] Кэмпбелл М. Будуюмо Web-сайти. Дизайн. HTML. CSS. М.: Триумф, 2006. 480 с.

элемента `style` застосувати `inline` стиль, тобто призначити стиль конкретному HTML-елементу безпосередньо в документі, за допомогою HTML-атрибуту `style`.

Приклад CSS файлу зображений на рис. 8.

```
1  @charset "utf-8";
2  * {
3      margin: 0;
4      padding: 0;
5      outline: none;
6  }
7
8  html {height: 100%;}
9
10 body {
11
12     width: 100%;
13     height: 100%;
14     color: #333;
15     background: #fff;
16     font-size: 1em;
17     font-family: "Segoe IU", sans-serif;
18     line-height: 135%;
19
20 }
21
22
23 }
24
25 a {
26     color: #BDB76B;
27     text-decoration: none;
28     transition: all.6s ease;
29     -moz-transition: all.6s ease;
30     -webkit-transition: all.6s ease;
31     -ms-transition: all.6s ease;
32
33 }
34
35 a:hover {
36     color: #FF8C00;
37     text-decoration: none;
38     transition: all.6s ease;
39     -moz-transition: all.6s ease;
40     -webkit-transition: all.6s ease;
41     -ms-transition: all.6s ease;
42
43 }
44
45 a:active {color:#8ce4a6}
46
47 img {
48     max-width: 100%;
49     height: auto;
50     width: auto\9;
51 }
52
53 .left {float: left;}
54 .right {float: right;}
55
56 /* Блок для выделения*/
57
58 ::selection {background: #f58b8b; color:#fff}
59 ::-moz-selection {background: #f58b8b; color:#fff}
60 ::-webkit-selection {background: #f58b8b; color:#fff}
61 ::-ms-selection {background: #f58b8b; color:#fff}
62
63 header, footer {
```

Рисунок 8 – Приклад CSS документу

CSS діє іншим, зручнішим і економнішим способом [6]¹⁾. Для присвоєння якому-небудь елементу певних характеристик потрібно один раз описати цей елемент і визначити цей опис як стиль, а надалі просто вказувати, що елемент, який необхідно оформити відповідним чином, повинен прийняти властивості описаного стилю. Застосувати таблицю стилів до HTML-документу можна трьома способами: застосувати зовнішні стилі (у вигляді окремого текстового. CSS-файлу) за допомогою елементу `link` вбудувати стилі безпосередньо HTML документ (у вигляді блоку `css` тексту) за допомогою елемента `style` застосувати `inline` [7]²⁾ – стиль, тобто призначити стиль конкретному HTML-елементу безпосередньо в документі, за допомогою HTML-атрибуту `style`.

Препроцесор гіпертексту

PHP [8]³⁾ – це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом.

Синтаксис мови бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web-розробникам можливості швидкого створення динамічних web-сторінок.

Значною відзнакою PHP від іншого коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Користувач навіть може сконфігурувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипту.

Створення скриптів для виконання на стороні сервера. PHP найбільш широко використовується саме таким чином. Все, що знадобиться користувачу, це парсер PHP (у вигляді програми CGI або серверного модуля), веб-

¹⁾ [6] Ши Д. Філософія CSS дизайну. СПб.: НТ Пресс, 2015. 312 с.

²⁾ [7] Фрейн Б. HTML5 та CSS3. Розробка сайтів для будь-яких браузерів та пристроїв. СПб.: Питер, 2014. 304 с.

³⁾ [8] PHP – українське програмування. URL: <http://programming.in.ua/web-design/allphp.html> (дата звернення 03.05.2020).

сервер і браузер. Щоб користувач міг переглядати результати виконання PHP-скриптів в браузері, потрібен вебсервер і встановлений PHP.

Створення скриптів для виконання в командному рядку. Користувач може створити PHP-скрипт, здатний запускатися незалежно від веб-сервера та браузера. Все, що буде потрібно це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою утиліти cron [планувальник завдань](на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в задачах простої обробки текстів.

Створення додатків GUI (графічних інтерфейсів), що виконуються на стороні клієнта. Можливо, PHP є не найкращим мовою для створення подібних додатків, але, якщо користувач дуже добре знає PHP і може використати деякі його можливості у своїх клієнт-додатках, можна використовувати PHP-GTK для створення таких додатків.

PHP доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix (такі, як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, та багатьох інших. Існує навіть версія PHP для OS / 2.

Також в PHP включена підтримка більшості сучасних веб-серверів [9]¹⁾, таких, як Apache, Microsoft Internet Information Server (IIS), Personal Web Server, серверів Netscape і iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd та багатьох інших. Для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати як процесор CGI.

Приклад PHP файлу наведено нижче.

```
<?php
// Файл firstsql.php
```

¹⁾ [9] HTML CSS JavaScript MySQL PHP Bootstrap book academy. URL: <https://html5css.ru/> (дата звернення 09.05.2020).

```

$host='localhost'; // ім'я хосту (з'ясовується у провайдера)
$database='db_name';
// ім'я бази даних, яку необхідно заздалегідь створити
$user='user_name';
// задане ім'я користувачу, або визначене провайдером
$pswd='your_pass'; // заданий вами пароль
$dbh = mysql_connect($host, $user, $pswd)
or die("Не можу з'єднатися з MySQL.");
mysql_select_db($database)
or die("Не можу під'єднатися до бази.");
$query = "SELECT * FROM `my_sql_table`";
$res = mysql_query($query);
while($row = mysql_fetch_array($res)) {
echo "Номер: ".$row['id']."<br>\n";
echo "Ім'я: ".$row['firstname']."<br>\n";
echo "Прізвище: ".$row['surname']."<br><hr>\n"; }
?>

```

Опис середовища розробки

Основним середовищем розробки було середовище Sublime Text 3. Sublime Text – це швидкий текстовий редактор між платформами. Підтримуються плагіни, розроблені з використанням мови програмування Python. На рис. 9 представлено інтерфейс цього застосування.

Sublime Text не є безкоштовним чи відкритим програмним забезпеченням, але деякі плагіни розповсюджуються за допомогою безкоштовної ліцензії, розробленої та підтримуваної спільнотою розробників.

Дане середовище розробки має можливість завантажити плагін для необхідних мов програмування, який увімкне можливість розпізнавати теги або оператори окремих мов програмування для підвищення читабельності коду. Також буде увімкнена можливість автоматичного завершення слів.

```

1 <?php session_start(); ?>
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <meta name="keywords" content="text, site, website" />
8 <meta name="description" content="Справочник бармена" />
9 <link href="css/style.css" rel="stylesheet" type="text/css" />
10 <link href="images/favicon.ico" rel="shortcut icon" type="image/x-icon" />
11 <title>SHAKEBARMEN</title>
12 </head>
13 </html>
14
15
16
17 <body>
18 <div id="page-wrap">
19 <header>
20 <a href="index.html" title="На головну" id="logo">SHAKEBARMEN</a>
21 <span class="right">Home class="contact"</span>
22 <span class="contact"></span>
23
24 <?php include("../includes/userbox.php"); ?>
25
26 </header>
27 <div id="mainmenu">
28 <div id="menu">Меню</div>
29 </div>
30 <?php include("../includes/menu_items.txt"); ?>
31 </div>
32
33 <div id="wrapper">
34 <div id="articles">
35
36 <article>
37 
38 <h2>Мастер-класс</h2>
39 <p>Смотри как и работает показывает высший класс опытные бармены.</p>
40 <a href="article" title="Посмотреть">читать далее...</a>
41 </article>
42
43 <article>
44 
45 <h2>Учись вместе с нами</h2>
46 <p>Рецепты для создания коктейлей и т.д.</p>
47 <a href="rec.php" title="Посмотреть">читать далее...</a>
48 </article>
49
50 <article>
51 
52 <h2>Лучшие барты в Украине</h2>
53 <p>Лучше барты где вас обслужат по высшему качеству.</p>
54 <a href="bar.php" title="Посмотреть">читать далее...</a>
55 </article>
56 </div>
57 </div>
58 </body>
59 </html>

```

Рисунок 9 – Интерфейс Sublime Text 3

Аналіз програмного середовища для створення серверу

Open Server – це портативна програмна платформа, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань. Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, володіє потужними можливостями з адміністрування та налаштування компонентів.

Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

OpenServer включає в себе: Apache, Nginx, MySQL, MariaDB, MongoDB, PostgreSQL, PHP, PHPMyAdmin.

Аналіз середовища для керування базами даних

Продукт phpMyAdmin використовувався для адміністрування створеної бази даних (редагування бази даних).

PHPMyAdmin – це веб-додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СУБД MySQL. PHPMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL і переглядати вміст таблиць і баз даних. Dodatok користується великою популярністю у веб-розробників, так як дозволяє управляти СУБД MySQL без безпосереднього введення SQL команд, надаючи дружній інтерфейс.

На сьогоднішній день PHPMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, враховуючи всі нововведення СУБД MySQL. Більшість провайдерів використовують цю програму в якості панелі управління для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

4 РОЗРОБКА ДИЗАЙНУ WEB-САЙТУ

Так як розроблюваний сайт є інформаційним, то основний акцент потрібно робити на контенті. Тому дизайн сайту створено у мінімалістичному стилі, а саме з максимальним дотриманням його принципів. Мінімалізм – це стиль в дизайні, що характеризується лаконічністю виразних засобів, простою, точністю і ясністю композиції. Відкидаючи класичні методи та прийоми творчості і традиційні художні матеріали, мінімалісти використовують промислові та природні матеріали простих геометричних форм, нейтральних кольорів (чорний, сірий) і малих обсягів. Мінімалізм у веб-дизайні сьогодні дуже популярний. У цього стилю безліч переваг: користувач завжди гранично ясно розуміє де знаходиться і що робити, увагою відвідувачів легко управляти, а деякі елементи добре запам'ятовуються. Основним фактором мінімалізму у веб-дизайні є простий і зрозумілий інтерфейс, який не представляє різних стилів (градієнт, тінь). Головний принцип в мінімалізмі при створення веб-дизайну – це не більше трьох кольорів. Він також використовується в створенні іконок, дизайн сайтів, картинок.

Опис стилю мінімалізму

Збільшення швидкості завантаження сторінок [10]¹⁾ – це одна з головних причин вибору на користь мінімалізму, адже чим складнішим в плані дизайну є шаблон сайту, тим більше часу потрібно на завантаження сторінок в браузері відвідувача. Адже далеко не у всіх є доступ до високошвидкісного підключення, і такі люди не будуть чекати, поки сторінка завантажиться, вони просто повернуться до пошуку або перейдуть на інший сайт. Висока швидкість завантаження дає також позитивний SEO-ефект, причому, даний фактор ранжування був офіційно озвучений представниками пошукових систем.

Акуратний зовнішній вигляд. Чистий і акуратний дизайн більш привабливий, ніж перевантажений і нав'язливий. Це твердження вірне для

¹⁾ [10] П'ять причин використовувати мінімалістичний дизайн. URL: <http://www.imagecms.net/blog/obzory/minimalisticheskii-dizain-saita> (дата звернення 15.05.2020).

проектів будь-якого типу. Пропозиція використовувати різні ефекти в оформленні та безліч цікавих графічних елементів може бути привабливим, але, зазвичай це виглядає по дилетантськи, і призводить тільки до підвищення відсотка відмов.

Доступність на всіх пристроях. Оскільки все більше і більше людей виходять в Інтернет з мобільних пристроїв, необхідно думати і про зручність цих людей, працюючи над своїм сайтом. Ресурсом, який переобтяжений зайвим функціоналом, неможливо нормально користуватися з мобільного пристрою, і, в результаті, люди надовго не залишаються.

Простота у використанні. Складний у використанні сайт з безліччю відволікаючих чинників заважає домогтися від відвідувачів конверсійних дій. Оформлення в мінімалістичному стилі дозволяє підкреслити заклик до дії, і звернути увагу відвідувачів на комерційну пропозицію. Зрозуміла навігація і акцентування уваги на вмісті дозволяють людям зосередитися і не відволікатися від важливих речей.

Фокус на контент. Якість будь-якого сайту залежить від його змісту. При використанні мінімалістичного дизайну ніщо не повинно відволікати від контенту, читач повинен зосередити свою увагу на тому, що дійсно має значення. Сайт, який відрізняється високою якістю контенту, буде більш цінним як для відвідувачів, так і для роботи пошукових систем. «Прості» і «чисті» – це дві основні характеристики, які використовуються при описі дизайну сайтів, виконаного в стилі мінімалізму. Так як сайт використовує мінімалістичний стиль, то використано три кольори: білий (#ffffff), чорний (#000000) та сірий (#555555). У білому кольорі виконано фон сайту та пункти меню каталогу. Приклад наведений на рис. 10.

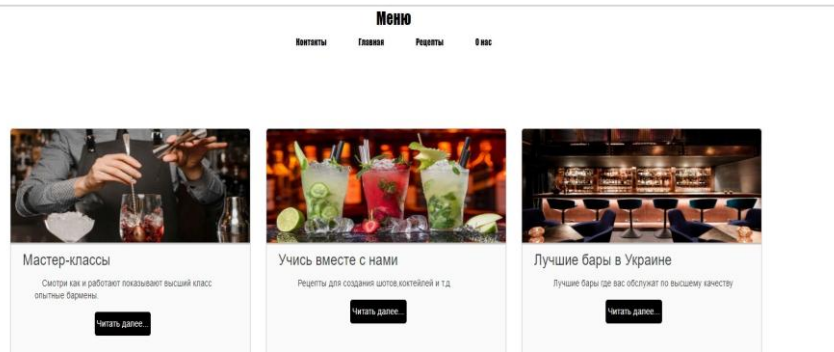


Рисунок 10 – Головна сторінка сайту

Чорний та червоний колір при наведенні використовується для верхнього горизонтального меню. Приклад наведений на рис. 11.



Рисунок 11 – Горизонтальне меню

Головне меню складається з горизонтального списку, що виконаний у чорному кольорі, що показано на рис. 11.

Головне меню складається з таких розділів: “Контакты, Главная, Рецепты, О нас”.

На головній сторінці розташовано 3 інформаційних блока з гіперпосиланнями (кнопками) для перегляду. Інформаційні блоки представлені на рис. 12.



Рисунок 12 – Інформаційний блок

Сторінка “Учись вместе с нами” розроблена у єдиному стилі. Контент розташовується у вигляді невеликих блоків, розділених лініями. На цій сторінці описується коктейлі, та їх склад. Приклад цієї сторінки представлений на рис. 13.

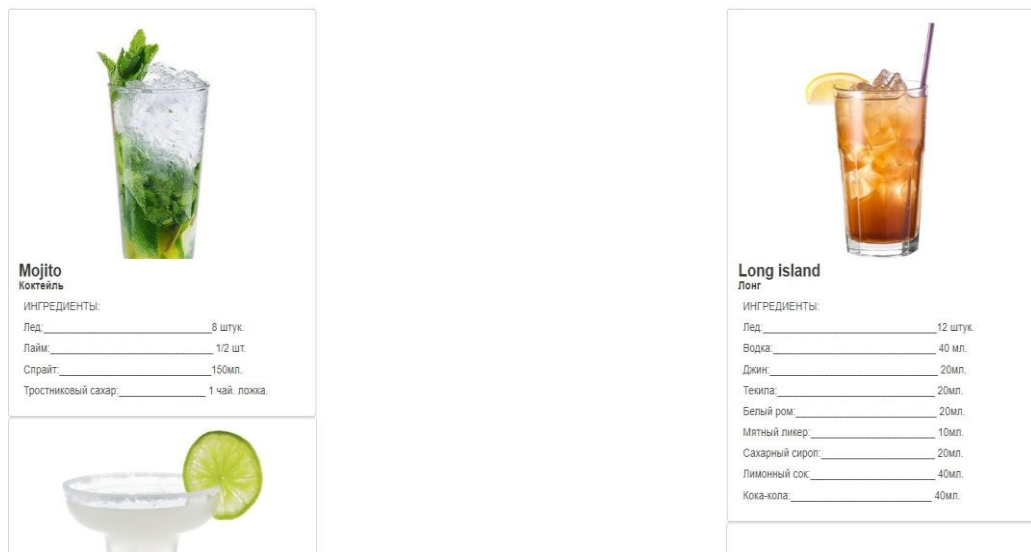


Рисунок 13 – Сторінка “Учись вместе с нами”

На сторінці “Лучшие бары в Украине” надані дані про бари та кафе. Їх адреса, графік роботи, контакти. Контент розташовується у вигляді невели-

ких блоків для зручного використання. Дану сторінку представлено на рис. 14.

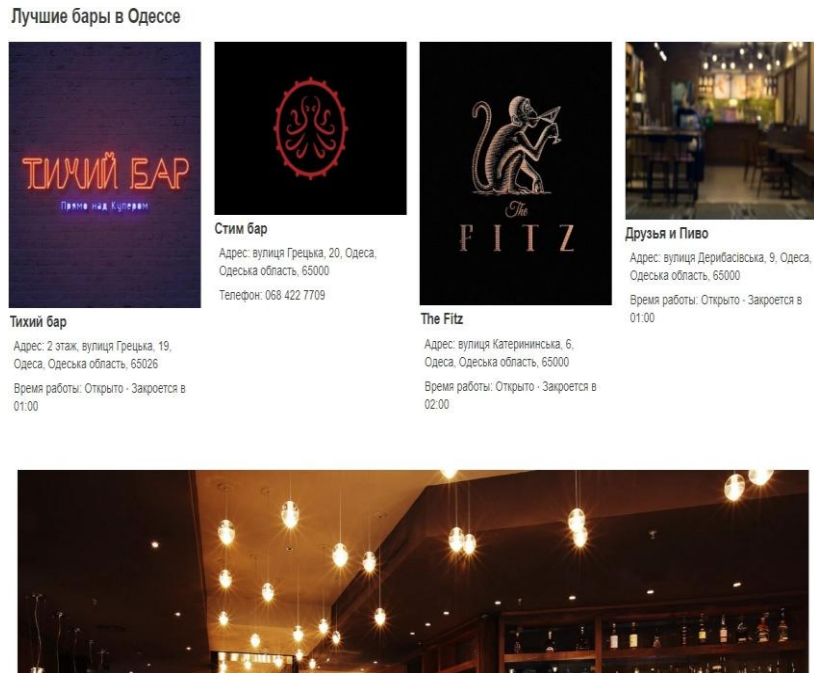


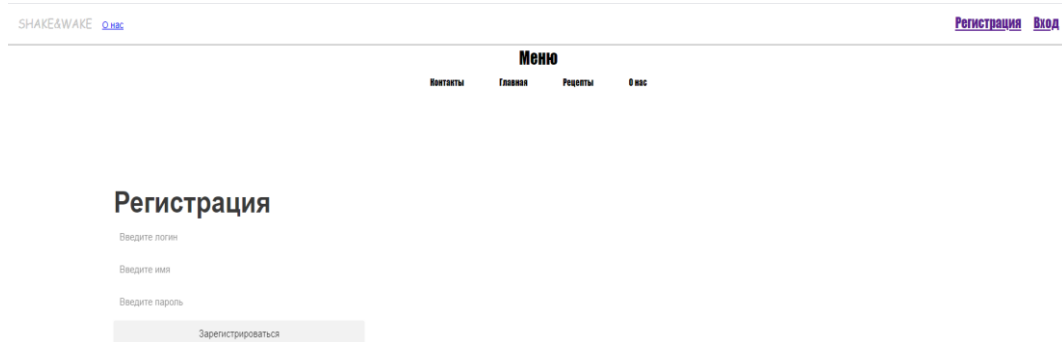
Рисунок 14 – Сторінка “Лучшие бары в Украине”

На сторінці “Контакты”. Будуть всі необхідні контакти для клієнтів. Так само інтегровано прив’язку карти на якій вказані координати необхідного місця. Дана сторінка представлена на рис. 15.



Рисунок 15 – Сторінка “Контакты”

Також на головній сторінці присутня форма реєстрації та форма логіну. в реєстрацію необхідно занести такі дані: ім'я, логін та пароль. Форма реєстрації та авторизації представлена на рис. 16 та 17 відповідно.



The screenshot shows the registration form on the SHAKE&WAKE website. The header includes the site name 'SHAKE&WAKE' with a home icon, a navigation menu with 'Контакти', 'Головна', 'Рецепти', and 'О нас', and links for 'Регистрация' and 'Вход'. The main heading is 'Регистрация'. Below it are three input fields labeled 'Введите логин', 'Введите имя', and 'Введите пароль'. A 'Зарегистрироваться' button is positioned below the password field.

Рисунок 16 – Форма реєстрації



The screenshot shows the login form on the SHAKE&WAKE website. The header is identical to the registration page. The main heading is 'Войти'. Below it are two input fields labeled 'Username' and 'Password'. A 'Login' button is positioned below the password field. A link 'Забыли пароль?' is located below the login button.

Рисунок 17 – Форма авторизації

Реєстрація прив'язана до Базы Даних MySQL, яка керується за допомогою системи керування баз даних PhpMyAdmin. Дані користувача зберігаються у таблицях. Базу даних та головну сторінку системи керування базами даних наведено на рис. 18.

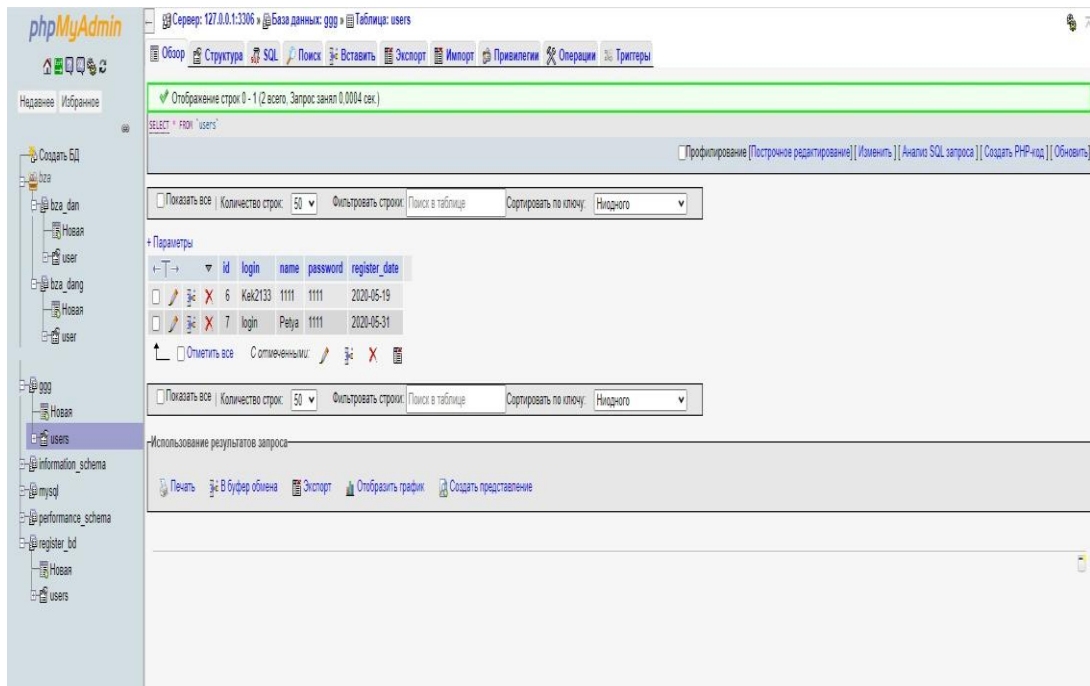


Рисунок 18 – База даних

Згідно з рис. 19. база даних має єдину таблицю для зберігання даних зареєстрованих користувачів. Оброблюється ім'я, пароль та системне ім'я користувача, яке є поштою. Кожен поштовий ящик має бути унікальним. Після авторизації у системі сайту його ім'я буде відображатися у правому верхньому кутку сайту.

Вы вошли как 1111 Выход

Рисунок 19 – Відображення користувача

Фізична структура сайту

Фізична структура сайту дозволяє змінювати дизайн за допомогою мови розмітки гіпертексту і каскадної таблиці стилів, і його функції за допомогою мов програмування. Сайт включає в себе наступні основні файли:

- Index.php – файл головної сторінки сайту, містить структуру розмітки головної сторінки;
- Bar.php – файл сторінки «Лучшие бары Украины», містить структуру розмітки елементів;
- About.php – файл сторінки «О нас», містить структуру розмітки елементів цієї сторінки;
- Contact.php – файл сторінки «Контакты», містить структуру розмітки елементів;
- Rec.php – файл сторінки «Рецепты», містить структуру розмітки елементів;
- Reg.php – файл сторінки «Регистрация», форма реєстрації;
- Autho.php – файл сторінки «Вход», форма входу;
- Loginner.php – файл для обробки входу;
- Style.css – файл містить таблиці стилів для настройки розташування елементів всіх сторінок сайту;
- каталог «images» – містить зображення, які використовуються для дизайну сайту;
- каталог «css» – містить файл “style.css”, який використовується для розташування елементів всіх сторінок сайту.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було створено дизайн та розроблено інформаційно-медійний сайт “SHAKE&WAKE”. Проаналізовано сучасні технології у створенні веб-сайтів. Найпопулярнішою тенденцією останніх років є мінімалістичний дизайн, який виключає нагромадження об’ємних об’єктів. Проаналізовано існуючі засоби для розробки веб-сайтів: порівняння серверних мов програмування PHP, HTML, CSS, різноманітних середовищ розробки та легких редакторів. Розроблено мережну структуру веб-сайту. Основні розділи веб-сайту: “Контакты”, “Рецепты”, “О нас”. Розроблено дизайн веб-сайту згідно з основними принципами мінімалістичного дизайну. Створено меню сайту. Розроблено горизонтальне меню каталогу та головне верхнє горизонтальне меню. На детальній сторінці виводиться інформаційні блоки з гіперпосиланнями. Також на сайті присутня форма реєстрації користувачів.

Протестовано веб-сайт у найпопулярніших браузерів. В ході тестування виявлено, що сайт працює коректно. Для готового програмного продукту було проведено повноцінне тестування: перевірка гіперпосилань, перевірка відповідності текстів заданій темі, перевірка відповідності сайту індивідуальному завданню, перевірка на кросбраузерність та юзабіліті. Отже, в ході виконання дипломної роботи було продемонстровано практичні навички створення та розробки веб-сайту.

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАННЯ

1. Глівенко С.В., Лапін Є.В., Павленко О.О. Інформаційні системи в менеджменті, навчальний посібник. ВТД «Університетська книга», Суми, 2003. 352 с.
2. Макарова М.В. Інформатика та комп'ютерна техніка, навчальний посібник. ВТД «Університетська книга», Суми 2003. 642 с.
3. Лабберс П. HTML5 для напівпрофесіоналів. Потужні інструменти для розробки сучасних веб-застосувань. М.: Вільямс, 2011. 272 с.
4. Джамса К. Ефективний самовчитель по креативному Web-дизайну. СПб: ДиаСофтЮп, 2005. 672 с.
5. Кэмпбел М. Будуємо Web-сайти. Дизайн. HTML. CSS. М.: Триумф, 2006. 480 с.
6. Ши Д. Філософія CSS дизайну. СПб.: НТ Пресс, 2015. 312 с.
7. Фрейн Б. HTML5 та CSS3. Розробка сайтів для будь-яких браузерів та пристроїв. СПб.: Питер, 2014. 304 с.
8. PHP – українське програмування. URL: <http://programming.in.ua/web-design/allphp.html> (дата звернення 03.05.2020).
9. HTML CSS JavaScript MySQL PHP Bootstrap book academy. URL: <https://html5css.ru/> (дата звернення 09.05.2020).
10. П'ять причин використовувати мінімалістичний дизайн. URL: <http://www.imagecms.net/blog/obzory/minimalisticheskii-dizain-saita>. (дата звернення 15.05.2020).

ДОДАТОК КОД ПРОГРАМИ

Лістинг файлу Logginer.php

```

<?php
session_start();
if ($_GET['logout']) {
    session_unset();
    session_destroy();
    header("Location: index.php");
    exit;
} else {
    $login = $_POST['login'];
    $password = $_POST['password'];
    $Mysqli = new mysqli ('localhost', 'root', 'root', 'piotr');

    $Query = "SELECT name FROM users WHERE login = '". $login. "'
and password = '". $password. "'";

    if ($result = $Mysqli->query($Query)) {
        if (mysqli_num_rows($result) <= 0)
        {
            header("Location: autho.php?Wrong=1");
        } else {
            $row = $result->fetch_assoc();
            $_SESSION['user'] = $row['name'];
            header("Location: index.php");
        }
    }
    exit; }
?>

```

Лістинг файлу rec.php

```

<?php session_start(); ?>
<!DOCTYPE html>
<html>
<head>
<meta name="keywords" content="test, site, website" />
<meta name="description" content="Справочник бармена" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<title>SHAKE&WAKE</title>
<style>
    </style>
</head>
<body>
    <div id="page-wrap">
        <header>

```

```

<a href="index.html" title="На главную"
id="logo">SHAKE&WAKE</a>
<?php include('./includes/userbox.php'); ?>
</header>
<div id="mainmenu">
  <div id="menu">Меню</div>
<ul>
  <?php include('./includes/menu_items.txt'); ?>
</ul>

</div>
<div id="wrapper">
  <div id="articles">
    <div id="recepti">

      <div id="columRight">
        <div style="text-align:center;"></div>
        <h2>Long island</h2>
        <h4>Лонг</h4>
        <p> ИНГРЕДИЕНТЫ: </p>

<p>Лед:_____12 штук.</p>
        <p>Водка:_____40 мл.</p>
        <p>Джин:_____20мл.</p>
        <p>Текила:_____20мл.</p>
        <p>Белый ром:_____20мл.</p>
        <p>Мятный ликер:_____10мл.</p>
        <p>Сахарный сироп:_____20мл.</p>
        <p>Лимонный сок:_____40мл.</p>
        <p>Кока-кола:_____40мл.</p>
      </div>

<div id="columRight_a">
<div style="text-align:center;">
</div>
<h2>Aperol Spritz</h2>
<h4>Лонг</h4>
        <p> ИНГРЕДИЕНТЫ: </p>
        <p>Просекко:_____100 мл.</p>
        <p>Газ.вода:_____50 мл.</p>
        <p>Цедра апельсина:_____по вкусу</p>
        <p>Аператив апероль:_____50 мл. </p>
</div>

<div id="columCenter">
  <div style="text-align:center;"></div>
  <h2>Mojito</h2>
  <h4>Коктейль</h4>
  <p> ИНГРЕДИЕНТЫ: </p>
  <p>Лед:_____8 штук.</p>
  <p>Лайм:_____1/2 шт.</p>
  <p>Спрайт:_____150мл.</p>

```

```

        <p>Тростниковый сахар:_____1 чай.
ложка.</p>
    </div>
    </div>
    <div id="columnCenter_a">
    <div style="text-align: center;"></div>
    <h2>Margarita</h2>
        <h4>Коктейль</h4>
    <p> ИНГРЕДИЕНТЫ: </p>
    <p>Текила:_____50мл.</p>
    <p>Апельсиновый ликер:_____20мл.</p>
    <p>Имбирный сироп:_____1 чай. ложка.</p>
<p>Лайм:_____1 штука.</p>
    </div>
    <div id="columnCenter_c">
    <div style="text-align: center;">
    <h2>Кола и грейпфрутовый ликер</h2>
    <h4>Лонг</h4>
    <p> ИНГРЕДИЕНТЫ: </p>
        <p>Грейпфрутовый ликер BOLS:_____50мл.</p>
        <p>Кока-кола:_____150мл.</p>
        <p>Грейпфрут:_____30 гр.</p>
        <p>Лёд:_____180гр.</p>
    </div>
    </div>
    <footer>
    <span class="left">Все права защищены &copy; 2020</span>
    <span class="right">Социальные сети</span>
    </header>
    </footer>

</body>
</html>

```

Лістинг файлу reg.php

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
    <meta name="keywords" content="test, site, website" />
    <meta name="description" content="Справочник бармена" />
    <link href="css/style.css" rel="stylesheet" type="text/css" />
    <link href="images/icon.ico" rel="shortcut icon" type="image/x-
icon" />

```

```

<title>SHAKE&WAKE</title>
<style>
</style>
</head>
<body>
<div id="page-wrap">
<header>
<a href="index.html" title="На главную" id="logo">SHAKE&WAKE<a/
><span class="contact"><a href="About.html" title="Информация о
нас">О нас</a></span>
<span class="right"><span class="contact"><a href="reg.html"
title="Зарегистрироваться">Регистрация</a>
</span>
<span class="contact"><a href="autho.html"
title="Войти">Вход</a>
</span>
</span>
</header>
<div id="mainmenu">
<div id="menu">Меню</div>
<ul>
<?php include('./includes/menu_items.txt'); ?>
</ul>

<div class="container mt-4">
<h1>Регистрация</h1>
<br>
<form action="check.php" method="post">
<input type="text" class="form-control" name="login"
id="login" placeholder="Введите логин"><br>

<input type="text" class="form-control" name="name"
id="name" placeholder="Введите имя"><br>

<input type="password" class="form-control" name="pass"
id="pass" placeholder="Введите пароль"><br>

<button class="btn btn-succes"
type="sumbit">Зарегистрироваться</button>
</form>
</div>
</div>

<footer>
<span class="left">Все права защищены &copy; 2020</span>
<span class="right">Социальные сети</span>
</footer>
</body></html>

```

Лістинг файлу index.php

```

<?php session_start(); ?>
<!DOCTYPE html>
<html><head>
<meta name="keywords" content="test, site, website" />
<meta name="description" content="Справочник бармена" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<title>SHAKE&WAKE</title>
<style></style></head><body><div id="page-wrap">
<header>
<span class="right"><span class="contact"></span>
<span class="contact"></span></span>
<?php include('./includes/userbox.php'); ?>
    </header>
    <div id="mainmenu">
<div id="menu">Меню</div>
    <ul><?php include('./includes/menu_items.txt'); ?>
    </ul>
        <div id="wrapper">
            <div id="articles">
<article>
<h2>Мастер-классы</h2>
<p>Смотри как и работают показывают высший класс опытные
бармены.</p>
<a href="/article" title="Посмотреть">Читать далее...</a>
</article><article>
<h2>Учись вместе с нами</h2>
<p>Рецепты для создания шотов,коктейлей и т.д </p>
<a href="rec.php" title="Посмотреть">Читать далее...</a>
</article><article>
<h2>Лучшие бары в Украине</h2>
<p>Лучшие бары где вас обслужат по высшему качеству</p>
<a href="Bar.php" title="Посмотреть">Читать далее...</a>
</article></div>
<?php
if (! isset($_SESSION['user'])) {
echo
'<div class="clear">'.
'<div id="main_soc_block">'.
'<div id="main_soc_block_in">'.
'<h3>Регистрация на сайте</h3>'.
'<p>Стань барменом своей мечты</p>'.
'<a href="reg.html" title="Регистрация">Регистрация</a>'.
'</div></div></div>';}
?></div><footer>
<span class="left">Все права защищены &copy; 2020</span>
<span class="right">Социальные сети
</span></footer></body></html>
```

Лістинг файлу autho.php

```
<!DOCTYPE html>
<html><head>
<meta name="keywords" content="test, site, website" />
<meta name="description" content="Справочник бармена" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<title>SHAKE&WAKE</title>
<style></style> </head><body><div id="page-wrap">
<header><span class="contact">
<a href="About.html" title="Информация о нас">О нас</a></span>
<span class="right"><span class="contact">
<a href="reg.html" title="Зарегистрироваться">Регистрация</a>
</span>
<span class="contact">
<a href="autho.html" title="Войти">Вход</a>
</span></span></header>
<div id="mainmenu">
<div id="menu">Меню</div>
  <ul>
    <li><a href="Contact.html">Контакты</a></li>
    <li><a href="index.html">Главная</a></li>
    <li><a href="rec.html">Рецепты</a></li>
    <li><a href="About.html">О нас</a></li>
  </ul>
  <div class="container">
<form action="/logginer.php" method="post">
  <div class="row">
    <h2 style="text-align:center">Войти</h2>
    </div>
    <input type="text" name="login" id="login"
placeholder="Username" required>
    <input type="password" name="password" id="password"
placeholder="Password" required>
    <input type="submit" value="Login">
  <?php if ($_GET["Wrong"])
echo '<p>Неверная комбинация логин/пароль</p>'; ?>
  </div>
<div class="center-container">
  <div class="row">  <div class="col">
    </div>  <div class="col">
    <a href="#" style="color:black" class="btn">Забыли пароль?</a>
    </div> </form></div></div></div></div> </div> </form>
</div></footer> </body></html>
```

Лістинг файлу check.php

```
<!DOCTYPE html>
```

```

<html>
<head>
    <meta http-equiv="Content-Type"
content="text/html;charset=UTF-8" />
    <meta name="keywords" content="test, site, website" />
    <meta name="description" content="Справочник бармена" />
    <link href="css/style.css" rel="stylesheet" type="text/css" />
    <link href="images/icon.ico" rel="shortcut icon" type="image/x-
icon" />
<title>SHAKE&WAKE</title>
<style></style> </head>
<body>
    <div id="page-wrap">
        <header>
<a href="index.html" title="На главную" id="logo">SHAKE&WAKE
</a><span class="contact">
<a href="About.html" title="Информация о нас">О нас</a></span>
<span class="right"><span class="contact">
<a href="reg.html" title="Зарегистрироваться">Регистрация</a>
</span><span class="contact">
<a href="autho.html" title="Войти">Вход</a>
</span></span></header> <div id="mainmenu">
<div id="menu">Меню</div>
    <ul>
        <li><a href="Contact.html">Контакты</a></li>
        <li><a href="index.html">Главная</a></li>
        <li><a href="rec.html">Рецепты</a></li>
        <li><a href="About.html">О нас</a></li>
    </ul>
    <div class="container mt-4">
<?php
if (! $_POST["login"])
die('<p class="error">Необходимо ввести логин</p>');
if (! $_POST["name"])
die('<p class="error">Необходимо ввести имя</p>');
if (! $_POST['pass'])
die('<p class="error">Необходимо ввести пароль</p>');
    $login = filter_var(trim($_POST['login']),
    FILTER_SANITIZE_STRING);
    $name = filter_var(trim($_POST['name']),
    FILTER_SANITIZE_STRING);
    $pass = filter_var(trim($_POST['pass']),
    FILTER_SANITIZE_STRING);
    $Mysqli = new mysqli ( 'localhost', 'root', 'root', 'piotr');
    if ($mysqli->connect_error) {
        die ( '<p class="error">Помилка на стороні серверу.
Спробуйте пізніше ( ' . $mysqli-> connect_errno. ')'. $Mysqli->connect_error
.')</p>'); }
    $query = "INSERT INTO users(login,name,password) VALUES ("
    ."". $login. ", ". "" . $name. ", ". "" . $pass. "" . ')';
    if ($Mysqli->query($query)){
        echo '<p>Вы были успешно зарегистрированы.</p><br />';

```

```
}else{
    die('<p class="error">Пользователь с таким именем уже
    существует.</p>'); } $Mysqli->close(); ?>
</div> </div> <footer>
<span class="left">Все права защищены &copy; 2020</span>
<span class="right">Социальные сети


</span>
</footer> </body> </html> <br> <hr>
<p class="default">Для продовження реєстрації перевірте свою
поштову скриньку <?php echo $_GET["email"]; ?></p>
<br> <hr> </body> </html>
```