

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

НКЦ заочної освіти

Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка інформаційної системи меблевого магазину «Mebe1007»

Виконав студент 5 курсу групи КН-5
Напрямок 6.050101 комп'ютерні науки
Мігов Микола Леонідович

Керівник асистент
Бучинська Ірина Вікторівна

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

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

Одеса 2020

ЗМІСТ

Список умовних позначень та скорочень	6
Вступ	7
1 Основні поняття про середовище FIGMA	9
1.1 Можливості та особливості Figma	9
1.1.1 Розрахований на багато користувачів режим редагування	9
1.1.2 Опис хмарного сервера зберігання файлів та компонентів середовища Figma	11
1.1.3 Фрейми, сітки. Наявність історії версій в середовищі Figma	12
1.2.1 Переваги середовища Figma	13
2 Опис середовищ для розробки ВЕБ-інтерфейсів.....	16
2.1 Переваги та недоліки сучасних середовищ для розробки веб-інтерфейсів	16
2.2 Figma – сервіс для розробки веб-дизайну	21
3 Розробка дизайну головної сторінки меблевого магазину в середовищі FIGMA	23
3.1 Створення дизайну header	23
3.2 Розробка дизайну першого екрану «Слайдер»	24
3.3 Дизайн каталогу «Товари» та «Дизайн»	25
3.4 Розробка другого екрану «Колекція».....	26
3.5 Створення екрану «Ми можемо повністю облаштувати»	26
3.6 Дизайн четвертого екрану «Про “МЕВЕЛ 007”»	27
3.7 Створення дизайну footer	28
4 Створення БД «МЕВЕЛ007» за допомогою інструмента PhpMyAdmin	29
4.1 Інструмент для адміністрування БД – Workbench.....	32
4.2 Інструмент для адміністрування БД – Navicat.....	33
4.3 Інструмент для адміністрування БД – PHPMyAdmin	33
4.4 Інструмент для адміністрування БД – dbForge Studio for MySQL.....	34
4.5 Інструмент для адміністрування БД – HeidiSQL.....	35

	5
4.6 Інструмент для адміністрування БД – SQL Maestro для MySQL	35
4.7 Інструмент для адміністрування БД – EMS SQL Manager для MySQL	36
4.8 Інструмент для адміністрування БД – SQLyog.....	36
4.9 Інструмент для адміністрування БД – DBTools Manager	37
4.10 Інструмент для адміністрування БД – MyDB Studio	37
Висновки.....	50
Перелік джерел посилання	53
Додатки.....	54
Додаток А Виконання SQL-запитів до таблиці mebel007.customers	55
Додаток Б Виконання SQL-запитів до таблиці mebel007.products.....	57
Додаток В Виконання SQL-запитів до таблиці mebel007.orders	61

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- БД – база даних.
- ОС – операційна система.
- ПЗ – програмне забезпечення.
- ПК – персональний комп'ютер.
- Android – операційна система для смартфонів, планшетів, електронних книг, фітнес-браслетів, ноутбуків і т.д.
- CSS – Cascading Style Sheets (формальну мову опису зовнішнього вигляду документа, написаного з використанням мови розмітки).
- iOS – мобільна операційна система для смартфонів, електронних планшетів, носяться програвачів і деяких інших пристроїв, що розробляється і випускається американською компанією Apple.
- PDF – Portable Document Format (міжплатформений відкритий формат електронних документів, спочатку розроблений фірмою Adobe Systems з використанням низки можливостей мови PostScript).
- PSD – Photoshop Document (растровий формат зберігання графічної інформації, що використовує стиснення без втрат, створений спеціально для програми Adobe Photoshop і підтримує всі його можливості).
- UI – User Interface (дослівно «інтерфейс користувача»).
- UX – User Experience (дослівно «досвід користувача»).
- Web – інтернет простір.

ВСТУП

На сьогоднішній день існує чи мало середовищ для розробки, створення веб-інтерфейсу Photoshop, Adobe XD, Sketch, Figma та інші. Кожен з представлених прикладів, має свої переваги та недоліки. Дана кваліфікаційна робота бакалавра орієнтована на ознайомлення з основними характеристиками такого середовища, як Figma. А також розробки інтерфейсу меблевого магазину в даному середовищі.

Figma – це онлайн-сервіс, тобто програма, що працює в браузері (але є і десктопна версія). Для роботи в ній потрібно авторизуватися і мати постійне інтернет-підключення. Файли можна зберігати локально і при необхідності імпортувати в браузерну версію.

Якщо в команді розробника не більше двох дизайнерів і у дизайнера немає необхідності в бібліотеці компонентів, то можна абсолютно безкоштовно використовувати Figma для будь-якої кількості проектів.

Існує умовна безкоштовність, тобто можна користуватися Figma безкоштовно, з такими основними обмеженнями:

- максимум два редактора;
- максимум три проекти;
- історія версій зберігається 30 днів.

Є можливість заплатити за Figma, тільки при додаванні додаткових дизайнерів до проекту або при необхідності додаткових можливостей, таких як бібліотека компонентів або нескінченна історія версій файлів.

Мета кваліфікаційної дипломної роботи полягає в проектуванні і розробці інформаційно-рекламного сайту магазину «Mebel007», що займається реалізацією меблів, з метою розширення сфери її діяльності.

В ході проведення дослідницької роботи виявилися затребуваними наступні завдання дослідження, такі як: аналіз, аналогія, порівняння, класифікація. Передбачається, що створений сайт буде конкурентоздатною і

буде використовуватися фірмою в якості рекламного засобу, а також приверне нових покупців, клієнтів і підніме рейтинг фірми по місту і області.

Дана кваліфікаційна робота бакалавра, складається з 60 сторінок, 18 рисунків, 6 таблиць та 7 посилань.

1 ОСНОВНІ ПОНЯТТЯ ПРО СЕРЕДОВИЩЕ FIGMA

Онлайн сервіси – це сайти, що надають послуги різного характеру. Їх першочергове завдання – полегшити роботу і значно заощадити час. Подібні сайти успішно замінюють комп'ютерні програми, які найчастіше коштують дорого і вимагають спеціального налаштування. За допомогою них в інтернеті можна робити майже все – проводити грошові операції, дізнаватися останні новини, спілкуватися, зберігати інформацію, редагувати її, пересилати та публікувати. До речі, нерідко в інтернеті зустрічаються повністю безкоштовні онлайн сервіси, які не вимагають оплат за свої послуги [1]¹⁾.

Figma – це крос-платформний онлайн-сервіс для дизайнерів інтерфейсів і веб-розробників. Figma дозволяє розробляти інтерфейси в онлайн-додатку. У Figma є дві ключові особливості: доступ до макету прямо з вікна браузера і можливість спільної роботи над документами.

1.1 Можливості та особливості Figma

1.1.1 Розрахований на багато користувачів режим редагування

До появи Figma кільком дизайнерам було складно працювати над одним проектом, а також передавати макети розробникам. Photoshop відмовлявся коректно відкривати макет, до тих пір поки не встановити потрібні шрифти. Або виникали складності з парної роботою – колега вносив зміни в свою копію і забував сказати вам про це або помилково надсилав не ту версію макета.

Команда Figma врахувала всі ці проблеми і створила продукт, який з коробки включає в себе одночасну роботу кількох дизайнерів, версії

¹⁾ [1] Что такое ON-LINE сервисы. URL: <https://avada-media.ua/services/on-line-servisy/> (дата звернення: 21.03.2020)

документа і багато інших можливостей. Вони спрощують життя як дизайнеру, так і розробнику. Для того, щоб співпрацювати з іншими дизайнерами, в Figma передбачена опція створення команди (рис.1).

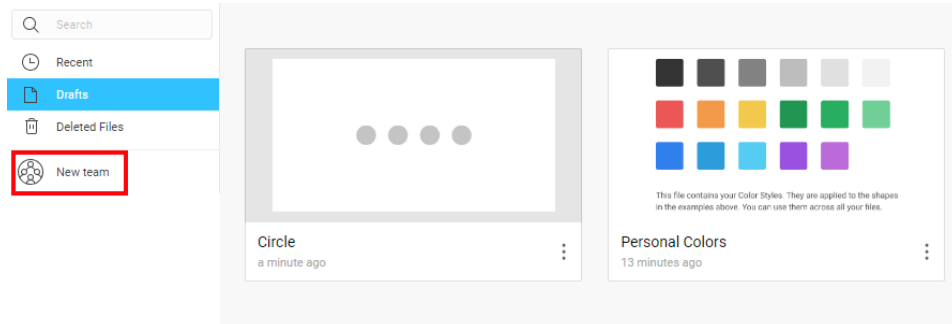


Рисунок 1 – Середовище Figma «Створення команди»

Все інтуїтивно зрозуміло. Потрібно вказати назву команди, вибрати потрібний тариф (див. рис.2).

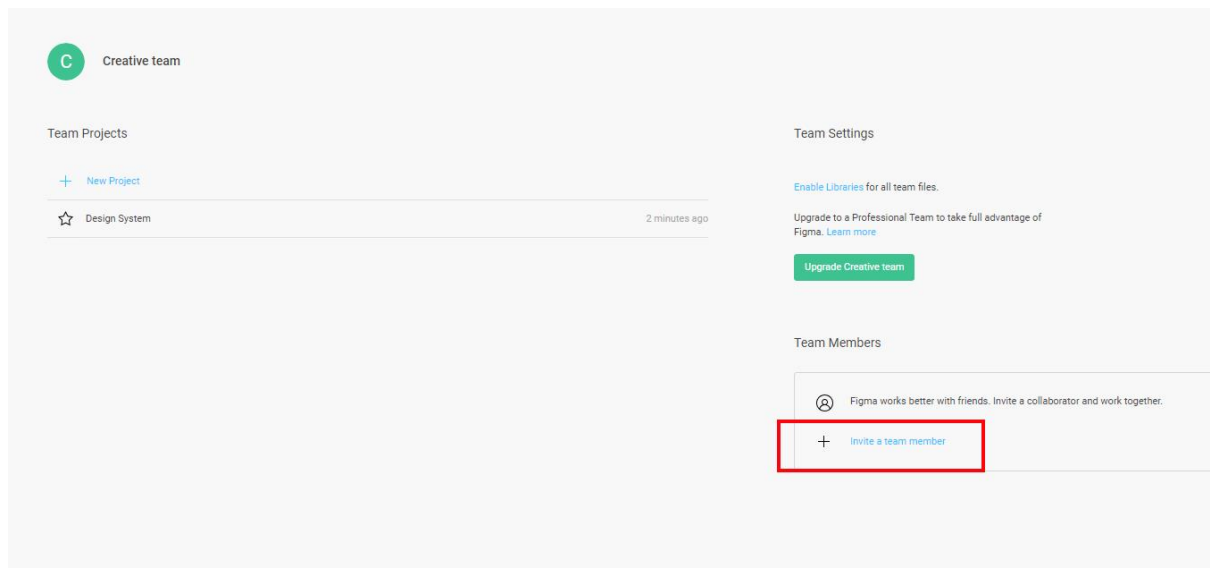


Рисунок 2 – Заповнення форми для створення команди в середовищі Figma

Після створення команди, залишається лише запросити її учасників і призначити рівень доступу до проекту (рис.3).

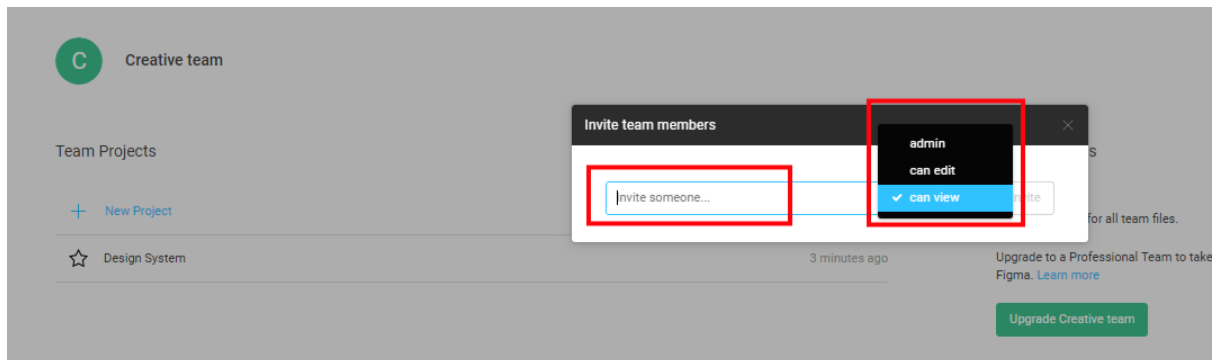


Рисунок 3 – Вікно присвоєння доступу до проекту учасникам команди в середовищі Figma

1.1.2 Опис хмарного сервера зберігання файлів та компонентів середовища Figma

Всі файли зберігаються в хмарі Figma. Не потрібно згадувати, де лежать макети, завантажувати і після внесення змін, завантажувати їх назад. Зміни в Figma теж зберігаються автоматично.

Компоненти – це елементи призначеного для користувача інтерфейсу, яким можна задавати загальні стилі – і потім швидко міняти їх у всьому макеті відразу.

Наприклад, після того, як була закінчена розробка 50 макетів в Photoshop, замовник попросив поміняти пункт меню в шапці і надіслати всі макети на затвердження. У середовищі Photoshop, потрібно відкривати файл за файлом і вносити зміни, як варіант, скопіювати шапку в окремий файл, внести зміни і продублювати в усі макети.

У Figma ця проблема вирішена: при зміні радіусу округлення однієї кнопки – змінюються стилі всіх елементів з цим стилем.

1.1.3 Фрейми, сітки. Наявність історії версій в середовищі Figma

Ця можливість дозволяє переглядати ранні версії файлу і відновлювати або дублювати будь-яку з них. Версії автоматично зберігаються, якщо протягом 30 хвилин у файлі не було ніяких змін.

Користувач може побачити, коли файл був відредагований і хто зробив зміни. Також є можливість вручну додавати нові версії при необхідності.

Фрейми у Figma, дуже схожі на артборди в Photoshop, але мають велику гнучкість в роботі. Фрейми добре масштабуються і їх з легкістю можна використовувати в інших фреймах. Для фреймів є встановлені розміри пристроїв: Phones, Tablet, Desktop, Watch, Paper, Social Media. Тому немає потреби тримати їх в голові і плутатися в розмірах.

Панель Grid Layout дозволяє швидко створювати будь-яку сітку. На одному макеті в проекті можна використовувати необмежену кількість сіток. Сітки швидко і гнучко налаштовуються і можуть бути як з фіксованою шириною, так і гумовими, на весь екран. При необхідності сітці можна легко поміняти колір [2]²⁾.

1.2 Огляд середовища Figma

Figma – це програма нового покоління для розробки інтерфейсів програм, веб-сайтів і мобільних додатків.

Не зовсім розумно порівнювати Figma з іншими програмами для розробки інтерфейсів. Перш за все тому, що Figma є хмарним веб-сервісом надає нативні програми для установки на Mac і Windows. Тобто є можливість нічого не встановлювати, а працювати через браузер. Якщо коротко, то Figma

²⁾ [2] Что такое Figma: возможности и принципы работы. URL: https://skillbox.ru/media/design/chto_takoe_figma/ (дата звернення: 23.03.2020)

це як Sketch, тільки без плагінів і з трохи іншою системою реалізації Символів і деяких інших функцій в кращу сторону.

Але ключовою особливістю є хмарна технологія, яка відкриває кардинально нові можливості для розробників та інших членів команди беруть участь в розробці продукту.

1.2.1 Переваги середовища Figma

Figma – це середовище, яке допоможе забути про зберігання файлів. За допомогою цієї програми, команда, яка працює над створення проекту, більше може не використовувати Dropbox, Google Drive і інші сервіси, за допомогою яких раніше виконувалася зберігання файлів проекту. Всі файли зберігаються в хмарі Figma. Також потрібно підкреслити, що в Figma відбувається автоматичне збереження змін.

Другою перевагою є збереження preview (попереднього переглянувши) зображення. При роботі в Photoshop, часто виникає ситуація, коли в проекті багато екранів, і для збереження конкретного екрану потрібно кожен раз виділяти артборди або ж відкривати окремий вихідний файл в Photoshop і окремо зберігати кожен макет або пропускати його через скрипт. У Figma у немає необхідності що-небудь зберігати. Можна просто відкрити доступ користувачеві поштою і дати йому посилання. Проект також з'явиться в його акаунті Figma. Або можна зробити проект відкритим для перегляду для всіх бажаючих за посиланням.

Наступною (третьою) перевагою, можна виділити «коментування макетів». Розглянемо, реалізація проекту, з точки зору дизайну в середовищі Photoshop. Процес достатньо складний:

- отримання текстового списку коментарів (недолік – не завжди зрозуміло про яке місце йдеться);
- написання коментар поверх зробленого скріншоту;
- конвертування макету з psd в pdf (і вже в pdf написання коментарів);

– використання dropbox або invision в якості сервісу коментування.

В середовищі Figma будь-який користувач, у якого є доступ до проекту може залишити коментар до потрібного місця.

До четвертої переваги, слід віднести те, що Invision, Marvel та інші сервіси, які використовувалися для прототипування раніше більше не потрібні. Оскільки функція прототипування вбудована в Figma за замовчуванням.

Наступна, п'ята перевага – це наявність режиму презентації. Під час того, як замовнику/користувачеві дають посилання на проект для перегляду в браузері або ж зі свого акаунту програми Figma; користувач може бачити те, що бачить дизайнер. Таким чином є можливість презентувати проект без необхідності використання стрімінгових відео-сервісів, таких як Skype і інші. Досить зателефонувати або ж написати через будь-який месенджер і дати посилання на презентацію проекту.

Набагато зручніша система роботи Символів в порівнянні з програмою Sketch. Що дозволяє в дуже наочній формі на панелі шарів успадковувати символи. А також змінювати будь-які властивості екземпляра: текст, включати і вимикати видимість шарів, змінювати розмір і головне стилі.

Потрібно визначити також те, що більше немає необхідності в сторінці Symbols як в Sketch. У Figma є панель, що знаходиться в одному ряду з панеллю шарів, яка відображає всі символи проекту і надає зручний пошук по імені символу.

До переваг Figma потрібно віднести можливість завантаження своїх символів в «Бібліотеку компонентів», де при необхідності є можливість оновлення, а також швидко знаходити потрібний символ за допомогою пошуку по імені символу. Дизайнер може використовувати збережені компоненти в будь-якому проекті.

Figma повністю безкоштовна для будь-яких учасників проекту: менеджерів, клієнтів, верстальників і програмістів в режимі перегляду. Це

означає, що члени команди зможуть бачити всі макети і презентації абсолютно безкоштовно.

Найбільшою перевагою перевагою для верстальників і програмістів в Figma – є спеціальна панель Code, на якій можна скопіювати CSS-стилі для Web, а також код-розмітки розташування і стилів для Android і iOS.

Figma інтегрується з Zeplin. А також файли Figma можна вбудовувати в файли Dropbox Paper.

В інших програмах це робиться або інструментом, які обмежує потреби або за допомогою напрямних, що вкрай незручно. У Figma є панель Grid Layout, аналогів якої поки не існує в інших програмах. За допомогою неї можна в кілька миттєвостей створити сітку будь-якого ступеня складності.

Можливості Grid Layout:

- створити необмежена кількість сіток на одному макеті;
- зробити сітки гумовими або статичними;
- розфарбувати кожну сітку в окремих колір.

Фрейми нова концепція простору для макетів ніж прості артборди. Фрейми можна вставляти один в одного. Таким чином в одному макеті можна використовувати кілька вкладених один в одного сіток [3]³⁾.

³⁾ [3] Обзор программы Figma. URL: <http://figmadesign.ru/1-0-0-obzor-figma.html> (дата звернення: 29.03.2020)

2 ОПИС СЕРЕДОВИЩ ДЛЯ РОЗРОБКИ ВЕБ-ІНТЕРФЕЙСІВ

Величезна кількість дизайнерів у всьому світі в якості інструменту для розробки веб-інтерфейсів використовує Adobe Photoshop. В цьому немає нічого дивного – Photoshop здатний впоратися практично з будь-яким завданням, яка має відношення до графіку.

Але як і всі комбайни, інструмент від Adobe в деяких випадках аж надто неповороткий і складний. При вирішенні специфічних завдань, включаючи розробку веб-інтерфейсів, можна спробувати і інші інструменти. Серед інших можна назвати Sketch, Adobe XD і Figma. Зараз пропонуємо поговорити саме про останньому, інші обговоримо в наступних публікаціях.

2.1 Переваги та недоліки сучасних середовищ для розробки веб-інтерфейсів

Відмінною особливістю Figma є те, що працювати з пакетом (повноцінно) можна в браузері. Figma також має і десктопну версію, але її функціональність лише трохи відрізняється від веб-сервісу (в десктопній версії можна працювати в офлайн, всі зміни автоматично будуть внесені при підключенні до мережі), тому можна нічого не завантажувати і не встановлювати на персональний комп'ютер.

Якщо ближче робота з десктопною версією, то її теж можна використовувати. Працює спритно та нарікань не викликає. Для роботи з Photoshop, найімовірніше, доведеться або провести апгрейд, або ж купити нову, більш потужну систему. А ось у випадку з Figma макети можна верстати хоч з планшета.

Графічні редактори в більшості випадків коштують грошей, і чималих. Figma – умовно-безкоштовне ПЗ. Присутня безкоштовна версія, можливостей якої цілком вистачить середньої руки дизайнеру. Безкоштовно можна

працювати не більше, ніж з трьома проектами і одним колегою-редактором (це стосується віддаленої спільної роботи). Історія версій зберігається не більше 30 днів. Користувачів з правами «read only» можна додавати скільки завгодно.

Якщо є потреба в більшій кількості можливостей, варто перейти на платну версію, яка обійдеться в 12 доларів в місяць. У ній є необмежена кількість проектів, «нескінченна історія», командна бібліотека елементів, з якої можна вибирати об'єкти в будь-який час, інтеграція з Slack і просунуті функції управління правами користувачів. За кожну додаткову редактора потрібно буде платити ще близько 12 доларів.

Що стосується різниці між редактором і користувачем, то в першому випадку ми говоримо фактично про адміністратора всіх проектів з повним доступом до всіх файлів в режимі читання і запису. У другому мова йде про користувача зі статусом «read only», якому, крім режиму перегляду, доступні ще коментування та деякі інші функції.

У Figma непогано організована спільна робота над окремими проектами рис.4.

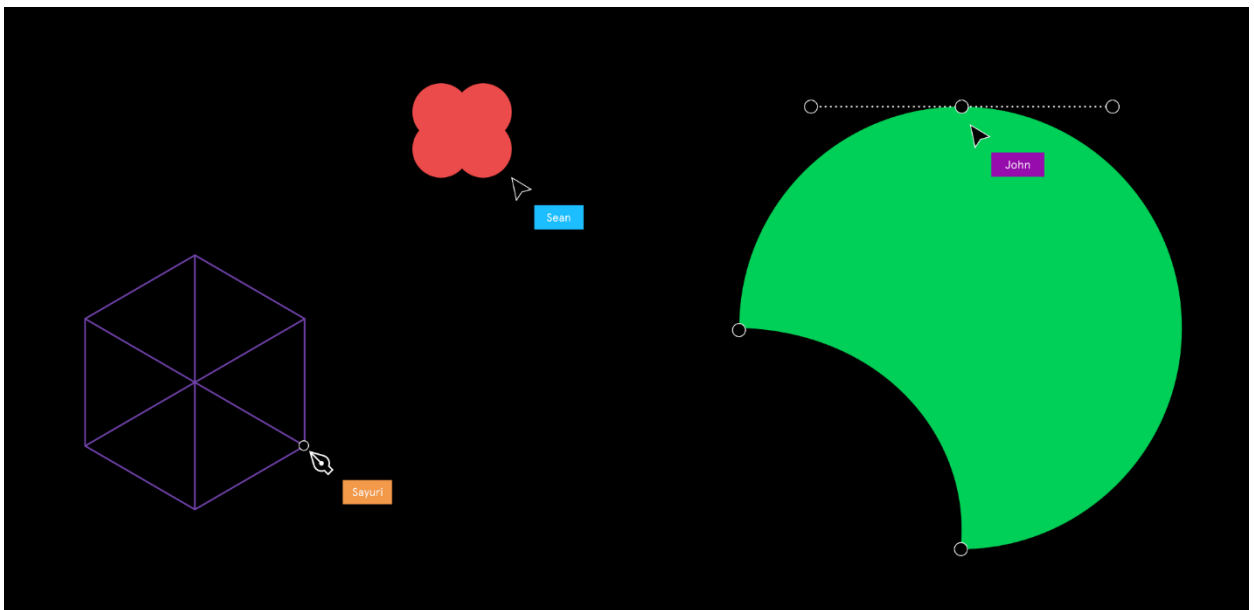


Рисунок 4 – Редагування проекту декількома учасникам команди в середовищі Figma

Як і у випадку з текстовими хмарними редакторами (той же Google Docs), курсори всіх користувачів проекту фарбуються різними кольорами і підписуються. Робота відстежується в режимі реального часу.

Крім цього, є можливість залишати коментарі. Їх можуть додавати не тільки редактори, але і звичайні користувачі.

Є інструменти комунікації, які дозволяють, не перериваючи роботи, обговорювати всі поточні питання в Figma. Наявність окремого каналу зв'язку дозволяє не відволікатися на сторонні комунікації на зразок телефону або скайпу.

Налаштування прав окремих користувачів потрібна для того, щоб ніхто не змінив те, що не потрібно, випадково чи ні. Крім того, в Figma є контроль версій – можна подивитися, хто і коли вносив зміни, а також відкотитися до потрібної версії всього в пару кліків в разі потреби див. рис.5.

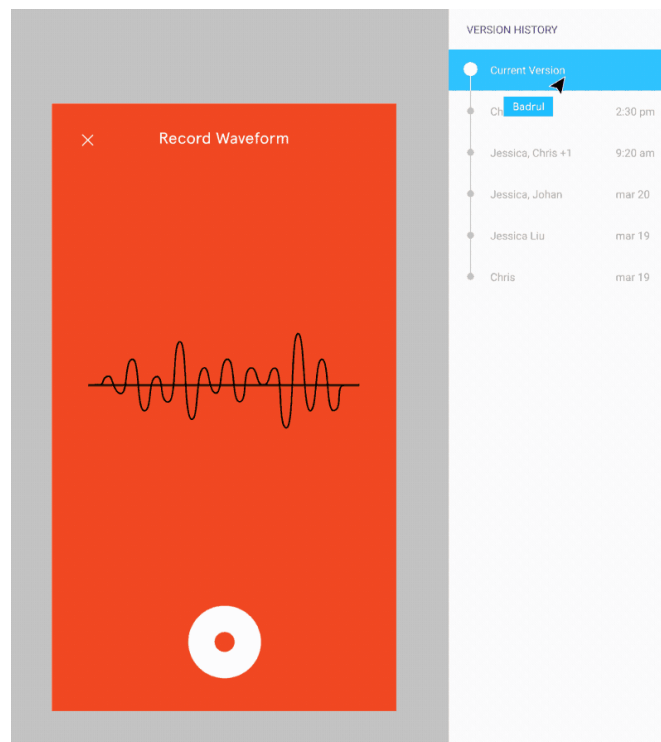


Рисунок 5 – Можливість контролю версіями проекту в середовищі Figma

Будь-який елемент створюваного інтерфейсу можна вибрати і при необхідності експортувати. Ця функція значно скорочує час, який потрібен на експорт того ж елемента в Photoshop.

При необхідності зберігається елемент можна збільшити в кілька разів. Ця функція може знадобитися при створенні веб-інтерфейсів для дисплеїв з підвищеною чіткістю.

Також в Figma є можливість створення залежних копій оригіналу. Залежність виражається наступним чином: якщо міняємо щось у копії оригінального елемента, ці зміни присвоюються тільки їй.

Але якщо змінюється початковий варіант елемента – оригінал, то зміни відбуваються у всіх інших копіях, де б вони не знаходилися.

У Figma є можливість пересувати замість векторних точок лінії. Функції не нова, вона з'явилася ще в 2016 році. Проте, при роботі з векторними сітками це дуже зручно.

У більшості векторних редакторів можна робити те ж саме, але часу йде більше, оскільки потрібно:

- виділити дві точки, які з'єднані лінією (вони дрібні, треба прицілитися як слід, і це не завжди виходить зробити з першого разу);
- з затиснутою кнопкою Shift зрушити їх в потрібну сторону (щоб не зламалася геометрія).

Що стосується Figma, то в середовищі додатка лінія виступає як би напрямляє для самої себе, потрібно просто захопити лінію курсором миші і відтягнути в напрямку, перпендикулярному самої лінії. Точки шукати не потрібно.

Перевагою Figma є те, що це середовище працює з Google Fonts (рис.6).

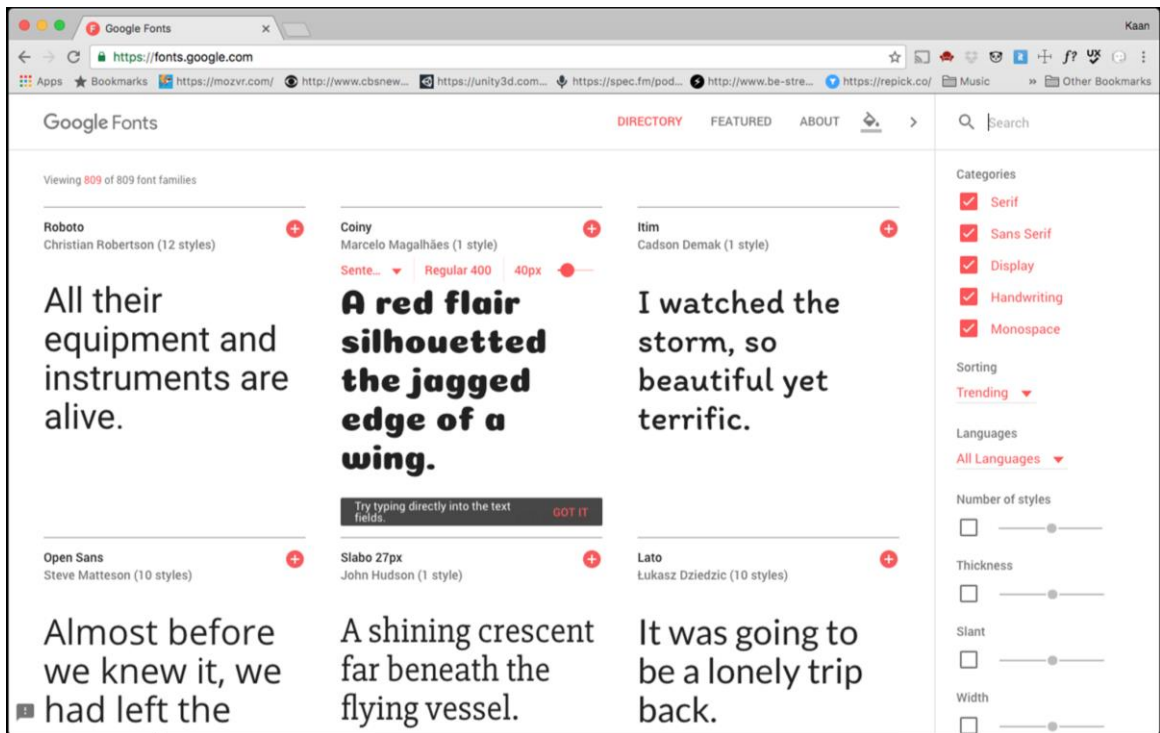


Рисунок 6 – Функція роботи з Google Fonts в середовищі Figma

Це означає, що більше не буде проблем з відсутніми шрифтами, через відсутність яких на ПК користувача, що відкриває макет, все шрифти можуть злетіти на дефолтні.

У Figma використовуються Google Fonts. Локальні шрифти при цьому завантажуються в проект.

Figma сумісна с файлами в форматі Sketch. Після імпорту робочі області, шари, групи зберігаються, включаючи властивості начебто кордонів і ефектів. Назви теж зберігаються. Імпортувати Sketch файли в Photoshop безпосередньо неможливо – для цього необхідно виконувати проміжні дії і використовувати додаткові програми.

Незважаючи на всі переваги середовища, Figma має свої мінуси:

- залежність від наявності підключення до мережі (якщо працювати в браузерній версії, то при відключенні інтернету файл, над яким велася робота, виявиться недоступним);
- відсутня округлення напів-пікселей;

- майже немає полігонів (раніше їх не було взагалі, зараз з'явилася можливість створення їх через Figma API).

Figma надає розробнику веб-інтерфейсів більшість життєво-важливих функцій. Середовище стає все більш популярним завдяки перерахованим вище інструментів і можливостям. Робота в "хмарі" – одна з головних фішок, яка властива поки що лише Figma.

Конкуренти редактора поки що не йдуть повністю в "хмари" і не зрозуміло, чи будуть переходити в найближчому майбутньому. У цьому є свої плюси, оскільки користувачі Photoshop та інших графічних редакторів можуть спокійно працювати в офлайні та не боятись відключення мережі [4]⁴⁾.

2.2 Figma – сервіс для розробки веб-дизайну

Обмеження сервісу Figma:

- не підходить для роботи з поліграфією (відсутній СМУК палітри і немає перемикачів розмірності сітки на см / мм);
- немає можливості безпосередньо імпортувати макети з Photoshop, для початку їх необхідно переводити в sketch формат.

Компоненти – спеціальний UI елементи в проектах, які повторюються. Компоненти можна створювати з кнопок, іконок, полів, меню, заголовків, форм і блоків. Виділяють дві групи компонентів – основні (Master components) і залежні (Instances). Тобто перша намальована кнопка буде основною, а якщо її скопіювати та зробити її компонентом, то друга вже залежна. І зміни першої підтягнуться для зміни другої кнопки. Це дуже зручна функція, яка максимально часто застосовується в сервісі Figma. За допомогою векторної сітки можна розробляти складні векторні форми. Лінії

⁴⁾ [4] Чем Figma лучше Photoshop для разработки веб-интерфейсов. URL: <https://habr.com/ru/company/skillbox/blog/359258/> (дата звернення: 12.04.2020)

в Figma не обов'язково повинні замикатися або виходити з однієї точки. Лінії можна розгалужувати і малювати звідки необхідно.

Відстеження історії дій кожного користувача, відновлення попередніх версій документів є в Figma. Особливість сервісу в тому, що він автоматично зберігає версії, якщо користувач закрив вкладку проекту або останні 30 хв. не вносив ніяких змін. Всі версії проекту можна відновлювати і дублювати. Безкоштовна версія – це зберігання за останні 30 днів. У платній версії доступна вся історія проекту з моменту його створення [5]⁵⁾.

⁵⁾ [5] Возможности Figma. Сервис для разработки веб-дизайна. URL: <https://hyperhost.ua/info/vozmozhnosti-figma-obzor-instrumenta-dlya-sozdaniya-veb-dizayna/> (дата звернення: 19.04.2020)

3 РОЗРОБКА ДИЗАЙНУ ГОЛОВНОЇ СТОРІНКИ МЕБЛЕВОГО МАГАЗИНУ В СЕРЕДОВИЩІ FIGMA

Створення дизайну для сайту або веб-застосунку – це самий суб'єктивно оцінюваний етап розробки, який часто викликає складності як на етапі постановки завдання, так і на етапі здачі-приймання виконаних робіт.

Дизайн – розробка графічних макетів інтерфейсу. Результат робіт по етапу: набір макетів основних екранів в форматі PSD, AI або в іншому придатному форматі, які потрібні для подальшої HTML-верстки. Іноді, крім основних екранів, також відмальовують окремі елементи інтерфейсу і складаються гайдлайни.

Для розробки дизайну меблевого магазину, було використано середовище Figma. Основними перевагами програми Figma:

- спільна робота над документами в режимі реального часу;
- зберігання інформації в віддаленому "хмарі" і можливість доступу до неї з будь-якої точки;
- windows, mac, linux підтримують роботу з даним сервісом;
- велика кількість корисного функціоналу, включаючи інтерактивні прототипи, векторні сітки, компоненти і багато іншого корисного для роботи в сфері веб-дизайну.

3.1 Створення дизайну header

Хедер (англ. Header – заголовок) – це верхній блок будь-який веб-сторінки, який називається також шапкою сайту і знаходиться протилежно футера (англ. Footer – підвалу сайту).

Хедер виступає одним з ключових елементів оформлення сайту. Він має прямий вплив на зовнішню привабливість ресурсу і його юзабіліті (англ. usability – зручність використання). Шапка грає важливу роль при оптимізації

сайту, оскільки саме в хедері розміщені ключові відомості про ресурс. Оптимізація сайту – це комплекс методів, який зробить сайт не тільки цікавим для користувача, але й зрозумілим та близьким для пошукових систем.

Завдання хедера давати користувачам відповіді на основні питання: що за бренд представлений, які товари і послуги пропонуються користувачам, як зв'язатися з співробітниками компанії, чи є актуальні акції, які надаються гарантії. При просуванні інтернет-магазину також рекомендується розміщувати форми зворотного зв'язку або кнопку замовити дзвінок. Все інформацію повинна бути в текстовому вигляді.

На рис.7 представлений дизайн header. Якщо поглянути, ми можемо побачити, що в даній шапці сайту є вибір мови сайту, контактний номер телефону, рядок пошуку, вкладки «Мої бажання» і «Увійти», а також підключення до соціальних мереж, для того щоб слідкувати за останніми новинками та акціями компанії.

Нижче зліва, представлений логотип компанії, праворуч від логотипа знаходить навігаційне меню, після якого розташовується «Кошик покупок».



Рисунок 7 – Приклад дизайну header

3.2 Розробка дизайну першого екрану «Слайдер»

Далі переходимо до розробки дизайну слайдера. Слайдер – це спеціальний елемент веб-дизайну, який представляє собою блок певної ширини найчастіше в шапці веб-сторінки. Головна його фішка в умовах, що змінюються в ручному або автоматичному режимі елементах – картинок, текстів і посилань див.рис.8.



Рисунок 8 – Приклад дизайну екрану «Слайдер»

В наш час «слайдер», можна побачити на будь-якому сайті. В незалежності від призначення сайту. Існує дуже багато варіантів представлення слайдера. На даному сайті, був застосований один з простих слайдерів, на якому будуть змінюватися елементи в автоматичному режимі.

3.3 Дизайн каталогу «Товари» та «Дизайн»

Після створення «Слайдер», переходимо до розробки каталогу. На головній сторінці для даного сайту, був створений дизайн для невеликого каталогу. Також потрібно зауважити, що крім вибору категорії товару (дивани, крісла, стільці і т.д.) є можливість підібрати дизайн (рис.9).



Рисунок 9 – Приклад дизайну екрану «Товари», «Дизайн»

3.4 Розробка другого екрану «Колекція»

На другому екрані «Колекції», достатньо простий дизайн. Зверху вказана назва блоку/екрану. Після якого, у вигляді карток представлені категорії товарів з нової колекції див.рис.10.

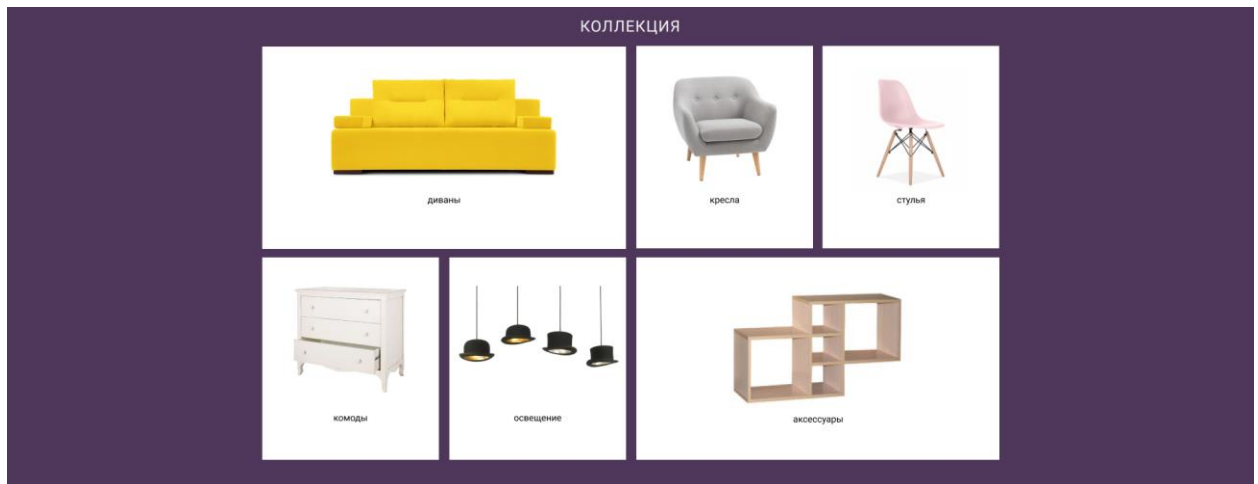


Рисунок 10 – Приклад дизайну екрану «Колекція»

3.5 Створення екрану «Ми можемо повністю облаштувати»

Далі, після створення дизайну для екрану «Колекція» переходимо до створення дизайну третього екрану «Ми можемо повністю облаштувати». В даному блоці, представлені варіанти облаштування кухні, гостьової кімнати, спальні. Нижче на рис.11, представлений дизайн карток, який складається з переднього за заднього фону.

МЫ МОЖЕМ ПОЛНОСТЬЮ ОБУСТРОИТЬ



Рисунок 11 – Приклад дизайну екрану «Ми можемо повністю облаштувати»

Потрібно зауважити, що передній та задній фон – це одна й та ж картинка. Тільки картинка на задньому фоні затемнена та має підпис. Якщо ж дивитися на передній фон, то картинка менших розмірів та виступає за картинку на задньому фоні, що при скролінгу сайту, повинно звернути увагу користувача.

3.6 Дизайн четвертого екрану «Про “МЕВЕЛ 007”»

На рис.12, представлений дизайн четвертого екрану «Про “МЕВЕЛ 007”». Як можна побачити з нижче представленою фотографією, ми маємо заголовок, після якого зліва знаходиться фотографія компанії, а праворуч – невеликий текст, який знайомить користувача з компанією.

Невеликий за розміром текст, приверне увагу користувача, а якщо відвідувач сайту, захоче прочитати більше інформації про компанію, він може натиснути на стрілку або текст(кнопку) «Читати далі» в нижньому

правому кутку – та перейти в розділ, де буде більше інформації про історію створення чи розвитку компанії.



Рисунок 12 – Приклад дизайну екрану «Про “МЕБЕЛ 007”»

3.7 Створення дизайну footer

І нарешті ми переходимо до створення дизайну footer. Футер (англ. Footer – підвал) – важливий структурний елемент будь-якого сайту, його часто називають «підвалом». Він розміщується в нижній частині веб-сторінки і є елементом, протилежним хедер (шапці сайту), яка знаходиться нагорі. Підвал виконує функцію додаткового блоку для розміщення даних компанії. Правильно заповнений футер добре впливає на просування сайту і покращує його юзабіліті. Зазвичай в підвалі розміщують дані, які потрібні на будь-якій сторінці і які завжди повинні бути «під рукою» користувача. Нерідко в підвалі дублюються відомості з хедера. Залежно від типу сайту визначається перелік даних для футера. Він формується в залежності від зовнішнього вигляду сторінок, змісту інших блоків і ін.

В нашому випадку, в футер частково дублюються елементи з хедера. А саме логотип, навігаційне меню та посилання на соціальні мережі (рис.13)



Рисунок 13 – Приклад дизайну footer

4 СТВОРЕННЯ БД «МЕВЕЛ007» ЗА ДОПОМОГОЮ ІНСТРУМЕНТА PHPMYADMIN

Під базою даних (БД) розуміють сховище структурованих даних, при цьому дані повинні бути несуперечливі, мінімально надлишкові і цілісні.

Реляційні БД представляють пов'язану між собою сукупність таблиць-сутностей бази даних (ТБД). Зв'язок між таблицями може знаходити своє відображення в структурі даних, а може тільки матися на увазі, тобто бути присутнім на неформалізованому рівні. Кожна таблиця БД представляється як сукупність рядків і стовпців, де рядки відповідають екземпляру об'єкта, конкретної події або явища, а стовпці – атрибутам (ознаками, характеристиками, параметрами) об'єкта, події, явища.

При практичній розробці БД таблиці-суті зводяться таблицями, рядка-екземпляри – записами, стовпці-атрибути – полями ТБД.

Одна з найважливіших переваг реляційних баз даних полягає в тому, що можна зберігати логічно згруповані дані в різних таблицях і задавати зв'язки між ними, об'єднуючи їх в єдину базу. Така організація даних дозволяє зменшити надмірність збережених даних, спрощує їх введення і організацію запитів і звітів.

У кожній таблиці БД може існувати первинний ключ. Під первинним ключем розуміють поле або набір полів, однозначно (унікально) ідентифікують запис. Первинний ключ повинен бути мінімально достатнім: в ньому не повинно бути полів, видалення яких з первинного ключа не відіб'ється на його унікальності.

Між двома або більше таблицями бази даних можуть існувати відносини підлеглості. Відносини підпорядкованості визначають, що для кожного запису головної таблиці {master, її називають іще батьківської} може існувати одна або кілька записів у підпорядкованій таблиці {detail, званої ще дочірньої}.

Існує три різновиди зв'язків між таблицями бази даних:

- «один-до-багатьох»;
- «один до одного»;
- «багато-до-багатьох».

Ставлення «один- до-багатьох» має місце, коли одному запису батьківської таблиці може відповідати кілька записів у дочірній таблиці.

Зв'язок "один- до-багатьох" є найпоширенішою для реляційних баз даних.

У широко поширеною нотації структури баз даних IDEF1X відношення «один-до-багатьох» зображується шляхом з'єднання таблиць лінією, яка на боці дочірньої таблиці закінчується кружком або іншим символом. Поля, що входять в первинний ключ для даної ТБД, завжди розташовані вгорі і відділяються від інших полів лінією.

Ставлення «один-до-одного» має місце, коли одному запису в батьківській таблиці відповідає один запис в дочірній таблиці.

Дане відношення використовують, якщо не хочуть, щоб таблиця БД «Не розпухала» від другорядної інформації.

Ставлення «багато-до-багатьох» має місце, коли:

- записи в батьківській таблиці може відповідати більше одного запису в дочірній таблиці;
- записи в дочірньої таблиці може відповідати більше одного запису в батьківській таблиці.

Багато СУБД (зокрема Access) не підтримують зв'язку «багато-до-багатьох» на рівні індексів і посилальної цілісності. Вважається, що будь-який зв'язок «багато-до-багатьох» можна замінити на одну або більше зв'язків «один- до-багатьох».

При проектуванні структури нової БД визначають сутність (об'єкти, явища) предметної області, які повинні знайти своє відображення в базі даних. В кінцевому підсумку аналіз предметної області повинен привести до створення ескізу БД. Спочатку бажано зобразити суті і зв'язку між ними. Як

правило, кожної суті в БД відповідає таблиця. Потім – в ескізі другого порядку – для кожної таблиці БД наводиться список атрибутів – полів записи.

На цьому етапі процес проектування структур БД є процесом творчим, неоднозначним, з іншого боку, вузлові його моменти можуть бути формалізовані.

Однією з таких формалізацій є вимога, згідно з яким реляційна база даних повинна бути нормалізована. Процес нормалізації має на меті усунення надмірності даних і полягає у приведенні до третьої нормальної формі (3НФ).

Перша нормальна форма (1НФ) вимагає, щоб кожне поле таблиці БД:

- було неподільним;
- не містило повторюваних груп.

Неподільність поля означає, що значення поля не повинно ділитися на більш дрібні значення. Наприклад, якщо в полі "Підрозділ" міститься назва факультету і назва кафедри, вимога неподільності не дотримується і необхідно з даного поля виділити або назва факультету, або кафедри в окреме поле.

Повторюваними є поля, що містять однакові за змістом значення. Наприклад, якщо потрібно отримати статистику здачі іспитів з предметів, можна створити поля для зберігання даних про оцінку по кожному предмету. Однак в цьому випадку ми маємо справу з повторюваними групами.

Друга нормальна форма (2НФ) вимагає, щоб всі поля таблиці залежали від первинного ключа, тобто, щоб первинний ключ однозначно визначав запис і не був надмірний. Ті поля, які залежать тільки від частини первинного ключа, повинні бути виділені в складі окремих таблиць.

Третя нормальна форма (ЗНФ) вимагає, щоб значення будь-якого поля таблиці, що не входить в первинний ключ, не залежало від значення іншого поля, що не входить в первинний ключ [6]⁶⁾.

В наш час створюють різні багатофункціональні програми для полегшення управління, розробки і адміністрування баз даних.

Більшість реляційних баз даних, за винятком MS Access, складаються з двох окремих компонентів: «back-end», де зберігаються дані і «front-end» – призначений для користувача інтерфейс для взаємодії з даними. Цей тип конструкції досить розумний, так як він распараллелює дворівневу модель програмування, яка відділяє шар даних від призначеного для користувача інтерфейсу і дозволяє сконцентрувати ринок ПЗ безпосередньо на поліпшенні своїх продуктів. Ця модель відкриває двері для третіх сторін, які створюють свої додатки для взаємодії з різними базами даних.

В Інтернеті можна знайти багато продуктів для розробки і адміністрування баз даних MySQL. Нижче представлено 10 найпопулярніших інструментів, які допоможуть заощадити час.

4.1 Інструмент для адміністрування БД – Workbench

Інструменту Workbench (розробка компанії Sun Systems / Oracle), який може працювати на платформах Microsoft Windows, Mac OS X і Linux. Workbench об'єднує в собі розробку й адміністрування баз даних і є наступником DBDesigner4.

MySQL Workbench поширюється під вільною ліцензією – Community Edition і з щорічної оплачуваної підпискою – Standard Edition. Остання включає в себе додаткові можливості, які здатні істотно поліпшити продуктивність, як розробників, так і адміністраторів баз даних.

Нижче представлений перелік, чому Workbench є популярним сьогодні:

⁶⁾ [6] Основные понятия БД. URL: <http://informatic.ugatu.ac.ru/lib/office/Proekt.htm> (дата звернення: 21.04.2020)

- можливість представити модель БД в графічному вигляді, а також редагування даних в таблиці;
- наявність простого і функціонального механізму по створенню зв'язків між полями таблиць, серед яких реалізована зв'язок «багато-до-багатьох» з можливістю створення таблиці зв'язків;
- функція Reverse Engineering дозволяє відновлювати структуру таблиць і зв'язків з тієї, яка була реалізована раніше і зберігається на сервері БД;
- наявність редактора SQL-запитів, який дає можливість при відправці на сервер отримувати відповідь в табличному вигляді і інші можливості.

4.2 Інструмент для адміністрування БД – Navicat

Друге місце займає Navicat (розробка компанії PremiumSoft CyberTech Ltd) – інструмент для розробки та адміністрування баз даних, який працює на будь-якому сервері MySQL, починаючи з версії 3.21. Для MySQL, Navicat доступний для роботи на платформах Microsoft Windows, Mac OS X і Linux.

Нижче представлений перелік, чому Navicat є популярним сьогодні:

- наявність візуального конструктора запитів;
- можливість імпорту, експорту та резервного копіювання даних;
- можливість створювати звіти;
- SSH і HTTP туннелінг;
- міграція та синхронізація даних і структури;
- інструмент для планування завдань і інші можливості.

4.3 Інструмент для адміністрування БД – РНРМуAdmin

РНРМуAdmin – безкоштовний додаток з відкритим кодом, призначене для адміністрування СУБД MySQL. РНРМуAdmin є веб-інтерфейс за

допомогою якого можна адмініструвати сервер MySQL, запускати команди і переглядати вміст таблиць і БД через браузер.

Нижче представлений перелік, чому PHPMyAdmin є популярним сьогодні:

- можливість управляти СУБД MySQL без безпосереднього введення SQL команд;
- як панель управління PHPMyAdmin надає можливість адміністрування виділених БД;
- інтенсивний розвиток;
- можливість інтегрувати PHPMyAdmin у власні розробки завдяки ліцензії GNU General Public License та інші можливості.

4.4 Інструмент для адміністрування БД – dbForge Studio for MySQL

dbForge Studio for MySQL – інструмент, який представляє інтерес як для користувачів MySQL, так і для розробників БД. З його допомогою ви зможете легко автоматизувати рутинну роботу і заощадити час. Сьогодні dbForge Studio for MySQL представлений в трьох редакціях: Express, Standard і Professional, що дозволяє вибрати той інструмент, який потрібен саме вам. Користуватися dbForge Studio for MySQL можна як комерційної, так і безкоштовною версією.

Нижче представлений перелік, чому dbForge Studio є популярним сьогодні:

- наявність коштів для централізованого адміністрування;
- інструменти для порівняння БД;
- візуальний профілювальник запитів;
- можливість управляти привілеями користувачів;
- наявність Дизайнера БД, який дозволяє будувати візуальні діаграми;
- поліпшена робота з проектами БД і інші можливості.

4.5 Інструмент для адміністрування БД – HeidiSQL

HeidiSQL – безкоштовний інструмент для управління базами даних. Гідна альтернатива PHPMyAdmin, яка дозволяє створювати і редагувати таблиці, уявлення, тригери, процедура, а також переглядати і редагувати дані. Також HeidiSQL надає можливість експорту даних як в SQL файл, так і в буфер обміну на інших серверах.

Нижче представлений перелік, чому HeidiSQL є популярним сьогодні:

- можливість підключатися до сервера за допомогою командного рядка;
- можливість пакетної оптимізації і відновлення таблиць;
- можливість редагування стовпців, індексів і зовнішніх ключів таблиць, редагування тіла і параметрів SQL процедур, тригерів і ін .;
- просте форматування неупорядкованих SQL;
- синхронізація таблиці між різними базами даних та інші можливості.

4.6 Інструмент для адміністрування БД – SQL Maestro для MySQL

SQL Maestro для MySQL – інструмент для адміністрування, розробки та управління найбільш затребуваних СУБД. Зручний графічний інтерфейс дає можливість виконувати SQL запити і скрипти, керувати привілеями користувачів, експортувати і створювати резервні копії даних.

Нижче представлений перелік, чому SQL Maestro є популярним сьогодні:

- підтримка версій MySQL сервера з версії 3.23;
- наявність конструктора баз даних;
- можливість редагування, угруповання, сортування та фільтрації даних;
- візуальний конструктор запитів;
- SSH і HTTP туннелінг;
- BLOB-редактор і інші можливості.

4.7 Інструмент для адміністрування БД – EMS SQL Manager для MySQL

EMS SQL Manager для MySQL – інструмент для розробки та адміністрування баз даних, який підтримує різні функції MySQL і працює з усіма версіями MySQL старше 3.23. З його допомогою у вас є можливість візуально редагувати, імпортувати і експортувати БД, виконувати сценарії SQL, управляти привілеями користувачів, візуально проектувати бази даних MySQL.

Нижче представлений перелік, чому EMS SQL Manager for MySQL є популярним сьогодні:

- підтримка даних UTF8;
- просте управління різними об'єктами MySQL;
- сумісність з усіма версіями з 3.23 по 6.0 включно;
- наявність графічних і текстових інструментів для формування запитів;
- SSH і HTTP туннелінг;
- зручний конструктор звітів і інші можливості.

4.8 Інструмент для адміністрування БД – SQLyog

SQLyog – один з найбільш потужних інструментів, який поєднує в собі можливості MySQL Administrator, PHPMyAdmin і деякі інші інструменти для адміністрування і розробки баз даних. SQLyog працює на платформах Microsoft Windows, Windows NT. і Linux за допомогою Wine.

Нижче представлений перелік, чому SQLyog популярним сьогодні:

- зручний конструктор запитів;
- можливість синхронізації даних;
- підтримка Unicode;
- SSH і HTTP, HTTPS тунелінг;
- «Розумне» автозавершення роботи;
- інтелектуальне додток коду і інші можливості.

4.9 Інструмент для адміністрування БД – DBTools Manager

DBTools Manager – додаток для управління даними, з вбудованою підтримкою MySQL, PostgreSQL, MSAccess, MSSQL Server, Oracle та інших БД. Підтримувані платформи: Windows 2000, XP, Vista, 7.

Нижче представлений перелік, чому DBTools Manager є популярним сьогодні:

- управління базами даних, таблицями;
- наявність редактора запитів;
- наявність майстра створення форм і звітів;
- можливість імпорту та експорту даних з різних джерел, серед яких MSAccess, MExcel, Paradox, FoxPro, DBF, ODBC таблиці, текстові і XML файли;
- конструктор діаграм і інші можливості.

4.10 Інструмент для адміністрування БД – MyDB Studio

MyDB Studio – безкоштовний інструмент для адміністрування БД MySQL, який дозволяє створювати, редагувати і видаляти записи, таблиці і бази даних. Працює виключно на платформі Windows.

Нижче представлений перелік, чому MyDB Studio є популярним сьогодні:

- можливість підключатися до необмеженої кількості баз даних;
- можливість підключення по SSH каналах;
- створення відкатів і експорт БД в різні формати;
- можливість перенесення, резервного копіювання, також відновлення БД і інші можливості [7]⁷⁾.

Для створення бази даних інтернет-магазину «Mebel007», було вирішено використовувати phpMyAdmin. На рис. 14 представлена структура БД Mebel007.

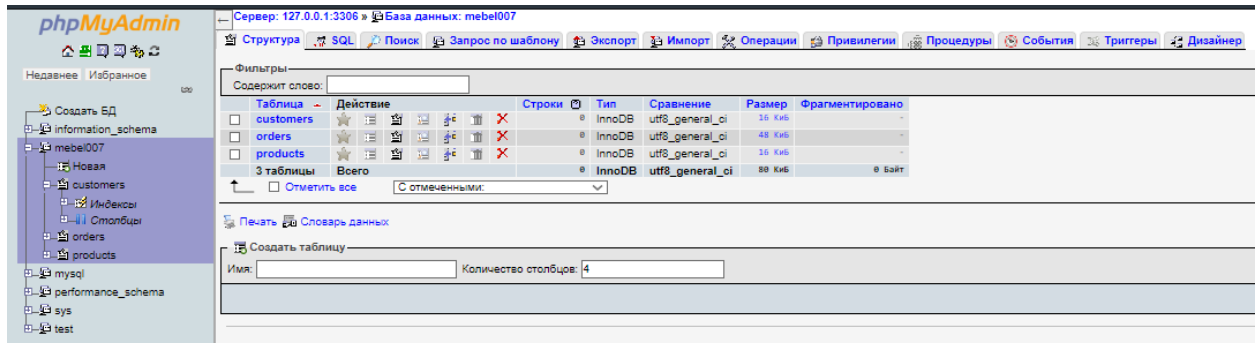


Рисунок 14 – Структура БД Mebel007

Як можна побачити з рис.14, БД mebel007 на даному етапі розробки БД має 3 таблиці, а саме «customers», «orders», «products». Почнемо зі створення таблиці «customers», вона представлена на рис. 15. Дана таблиця складається з 5 полів «CustomerID», «FirstName», «SecondName», «Phone», «Email».

⁷⁾ [7] 10 лучших инструментов для разработки и администрирования MySQL. URL: <https://habr.com/ru/post/142385/> (дата звернення: 01.05.2020)

Таблиця 1 – Структура таблиці «customers»

№	Ім'я	Тип	Порівняння	За замовчуванням
1	CustomerID	int(20)		Нет
2	FirstName	text	utf8_general_ci	NULL
3	SecondName	text	utf8_general_ci	NULL
4	Phone	varchar(24)	utf8_general_ci	NULL
5	Email	text	utf8_general_ci	NULL

Первинний ключ встановлений на полі «CustomerID» рис.15.

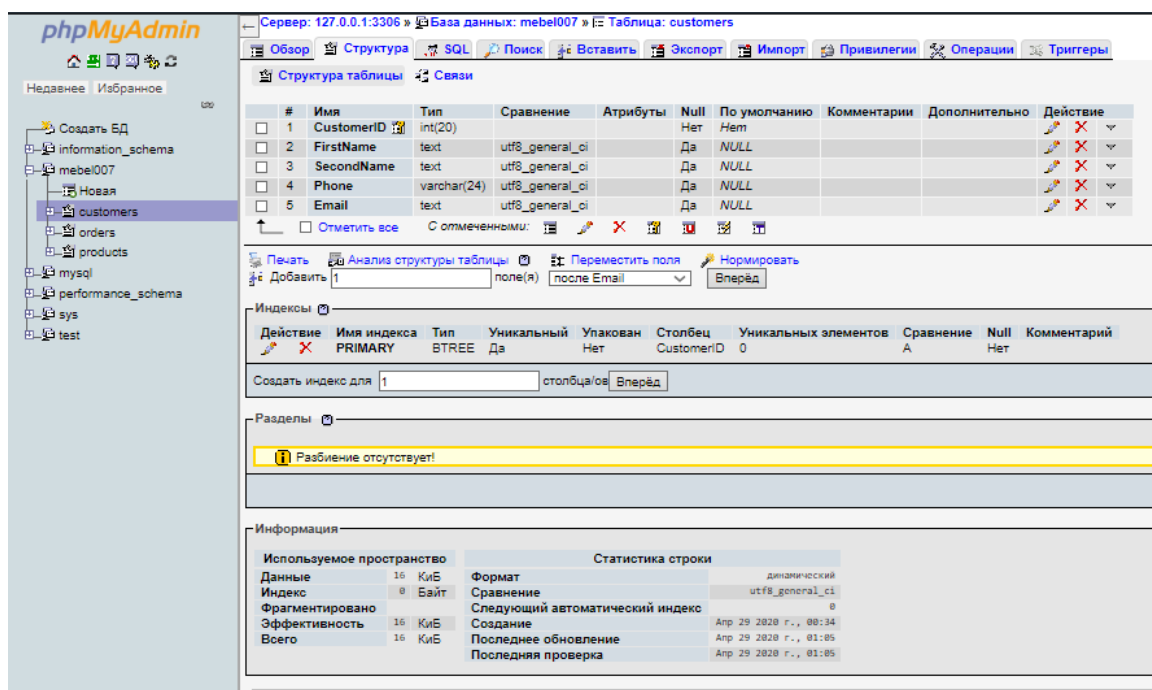


Рисунок 15 – Структура таблиці «customers»

Далі перейдемо до створення таблиці «products». Дана таблиця має 4 поля, а саме «ProductID», «ProductName», «Price».

Таблиця 2 – Структура таблиці «products»

№	Ім'я	Тип	Порівняння	За замовчуванням
1	ProductID	int(20)		Нет
2	ProductName	text	utf8_general_ci	NULL
3	Price	int(11)		NULL

4	TypeOfFurniture	text	utf8_general_ci	NULL
---	-----------------	------	-----------------	------

Первинний ключ встановлений на полі «ProductID» рис.16.

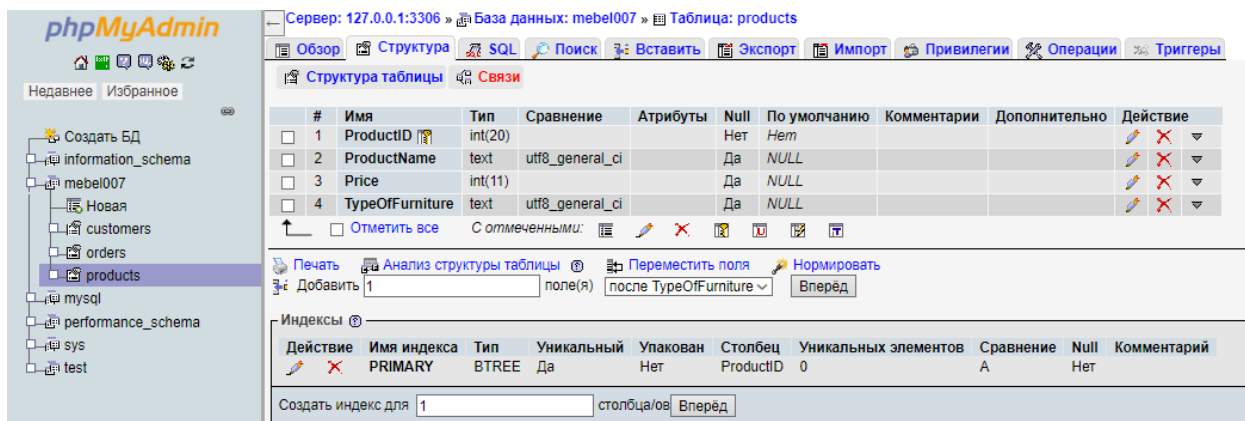


Рисунок 16 – Структура таблиці «products»

Наступним етапом є створення таблиці «orders». Таблиця «orders» містить в собі 5 полів, а саме «OrderID», «CustomerID», «ProductID», «OrderDate», «OrderQuantity».

Таблиця 3 – Структура таблиці «orders»

№	Ім'я	Тип	За замовчуванням
1	OrderID	int(11)	Нет
2	CustomerID	int(11)	Нет
3	ProductID	int(11)	NULL
4	OrderDate	date	NULL
5	OrderQuantity	int(11)	NULL

Первинний ключ встановлений на полі «OrderID» рис.17.

Сервер: 127.0.0.1:3306 » База данных: mebel007 » Таблица: orders

Структура таблицы

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	OrderID	int(11)			Нет	Нет			✕
2	CustomerID	int(11)			Нет	Нет			✕
3	ProductID	int(11)			Нет	Нет			✕
4	OrderDate	date			Да	NULL			✕
5	OrderQuantity	int(11)			Да	NULL			✕

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
✕	PRIMARY	BTREE	Да	Нет	OrderID	0	A	Нет	
✕	FK_Orders_Customers	BTREE	Нет	Нет	CustomerID	0	A	Нет	
✕	FK_Orders_Products	BTREE	Нет	Нет	ProductID	0	A	Нет	

Используемое пространство

Данные	Индекс	Фрагментировано	Эффективность	Всего
16 КиБ	32 КиБ	48 КиБ	48 КиБ	

Статистика строки

Формат	Сравнение	Следующий автоматический индекс	Создание	Последнее обновление	Последняя проверка
динамический	utf8_general_ci	0	Apr 29 2020 r., 08:58	Apr 29 2020 r., 01:05	Apr 29 2020 r., 01:05

Рисунок 17 – Структура таблиці «orders»

На рис.18 представлено зв'язки між таблицями БД mebel007.

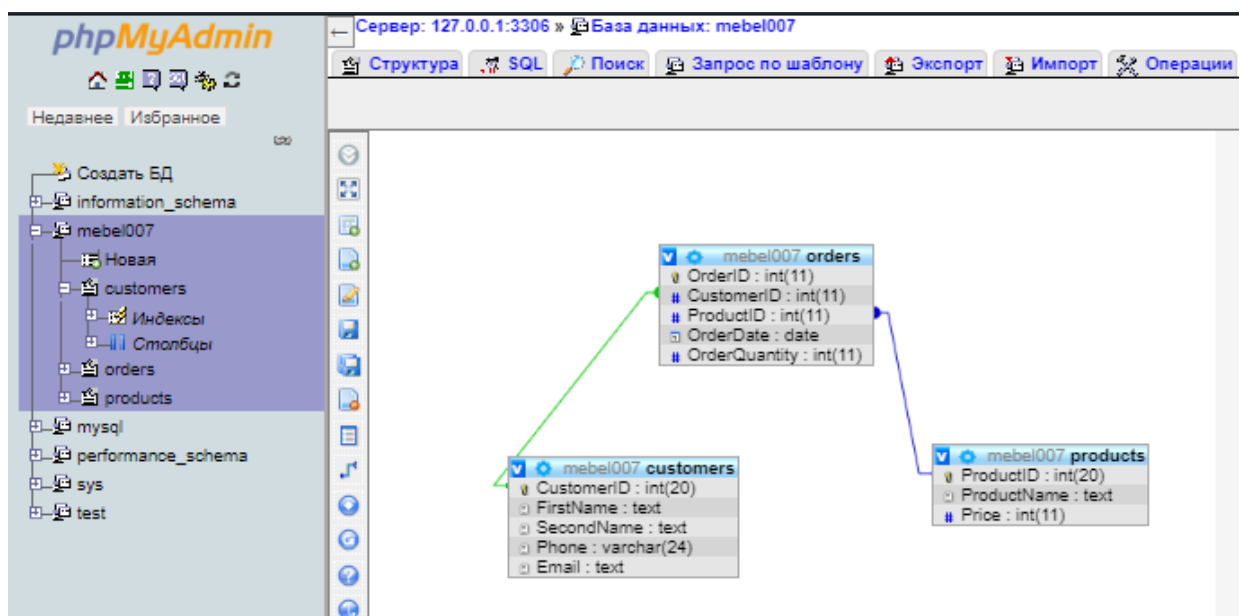


Рисунок 18 – Зв'язки між таблицями БД mebel007

Виконаємо SQL-запити до таблиці mebel007.customers :

```

-- Структура таблицы `customers`
CREATE TABLE `customers` (
  `CustomerID` int(20) NOT NULL,
  `FirstName` text,
  `SecondName` text,
  `Phone` varchar(24) DEFAULT NULL,
  `Email` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `customers`
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (1, 'Alexandr', 'Baver', '0670423123', 'baver63@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (2, 'Simona', 'Vaychute', '0639568963', 'simona_vaychute@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (3, 'Viktoriiia', 'Bodnar', '0950263741', 'bodnar-vik@mail.ru'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (4, 'Anastasia', 'Golovey', '0676793672', 'golovey20@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (5, 'Roman', 'Lykiyanenko', '(095)6767321', NULL),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (6, 'Igor', 'Babayev', '0973753147', 'babayev96@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (7, 'Victoria', 'Matunyk', '0632930532', 'vic.martunyk@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (8, 'Dasha', 'Kozuba', '0666578357', 'kozuba2017@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (9, 'Vitaliy', 'Veslov', '0935388321', 'veslovvvv1@mail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (10, 'Vladislav', 'Fomin', '0938787222', 'vlad.fomin@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (11, 'Masha', 'Lozuk', '0639393987', 'lozuk_masha@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (12, 'Ekaterina', 'Vistoma', '0503232369', 'vistoma_1993@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`, `Email`) VALUES (13, 'Egor', 'Krid', '0987417411', 'krid_krid123@mail.com'),

```



```

INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`,
`Email`) VALUES (14, 'Roma', 'Gorban', '0639551755', 'r_gorban@mail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`, `SecondName`, `Phone`,
`Email`) VALUES (15, 'Karina', 'Mileeva', '0673941555',
'mileeve.karina@gmail.com');

```

```

-- Индексы таблицы `customers`
ALTER TABLE `customers`
ADD PRIMARY KEY (`CustomerID`);
COMMIT;

```

Таблиця 4 – Дані таблиці «customers»

CustomerID	FirstName	SecondName	Phone	Email
1	Alexandr	Baver	0670423123	baver63@gmail.com
2	Simona	Vaychute	0639568963	simona_vaychute@ukr.net
3	Viktoriiia	Bodnar	0950263741	bodnar-vik@mail.ru
4	Anastasia	Golovey	0676793672	golovey20@ukr.net
5	Roman	Lykiyanenkko	(095)6767321	NULL
6	Igor	Babayev	0973753147	babayev96@gmail.com
7	Victoria	Matunyk	0632930532	vic.martunyk@gmail.com
8	Dasha	Kozuba	0666578357	kozuba2017@ukr.net
9	Vitaliy	Veselov	0935388321	veselovvvv1@mail.com
10	Vladislav	Fomin	0938787222	vlad.fomin@gmail.com
11	Masha	Lozuk	0639393987	lozuk_masha@ukr.net
12	Ekaterina	Vistoma	0503232369	vistoma_1993@ukr.net
13	Egor	Krid	0987417411	krid_krid123@mail.com
14	Roma	Gorban	0639551755	r_gorban@mail.com
15	Karina	Mileeva	0673941555	mileeve.karina@gmail.com

Виконаємо SQL-запити до таблиці mebel007.products :

```

-- Структура таблиці `products`
CREATE TABLE `products` (
`ProductID` int(20) NOT NULL,
`ProductName` text,
`Price` int(11) DEFAULT NULL,
`TypeOfFurniture` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- Дамп данных таблицы `products`

```

```

INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (1, 'EKTORP', 349, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (2, 'KIVIK', 69, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (3, 'BARNARP', 149, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (4, 'FRIHETEN', 339, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (5, 'BRUNSTA/HEMMA', 16, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (6, 'KNOPPARP', 79, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (7, 'RAGUNDA', 449, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (8, 'SLATORP', 799, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (9, 'ASARUM', 199, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (10, 'VIMLE', 849, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (11, 'NEIDEN', 75, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (12, 'SONGESAND', 261, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (13, 'DUNVIK', 824, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (14, 'SLAKT', 79, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (15, 'BRIMNES', 129, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (16, 'HEMNES', 299, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (17, 'ASKVOLL', 99, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (18, 'MALM', 249, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (19, 'NEIDEN', 39, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (20, 'SLATTUM', 99, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (21, 'RAKKETSTAD', 79, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (22, 'SONGESAND', 149, 'wardrobe'),

```

```

INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (23, 'PLATSA', 596, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (24, 'VUKU', 10, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (25, 'SMAGORA', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (26, 'KLEPPSRAD', 40, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (27, 'KVIKNE', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (28, 'BRIMNES', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (29, 'GODISHUS', 69, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (30, 'PAX', 256, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (31, 'FABYN', 45, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (32, 'LOBERGET', 25, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (33, 'LEIFARNE', 55, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (34, 'FLINTAN', 59, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (35, 'EKEDALEN', 45, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (36, 'TEODORES', 20, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (37, 'TERJE', 13, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (38, 'NILSONE/NORNA', 85, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (39, 'JANINGE', 29, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (40, 'GUNDE', 7, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (41, 'TORARED', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (42, 'MUSIK', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (43, 'TROSS', 8, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (44, 'SJO GANG', 9, 'lamps'),

```

```

INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (45, 'SINNERLIG', 50, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (46, 'LUFTNASSA', 13, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (47, 'EBBARED', 5, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (48, 'RANARP', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (49, 'MELODI', 4, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (50, 'KNISILINGE', 279, 'sofas');

```

```
-- Индексы таблицы `products`
```

```

ALTER TABLE `products`
ADD PRIMARY KEY (`ProductID`);
COMMIT;

```

Таблиця 5 – Дані таблиці «products»

ProductID	ProductName	Price	TypeOfFurniture
1	EKTORP	349	sofas
2	KIVIK	699	sofas
3	BARNARP	149	sofas
4	FRIHETEN	339	sofas
5	KNISILINGE	279	sofas
6	KNOPPARP	79	sofas
7	RAGUNDA	449	sofas
8	SLATORP	799	sofas
9	ASARUM	199	sofas
10	VIMLE	849	sofas
11	NEIDEN	75	beds
12	SONGESAND	261	beds
13	DUNVIK	824	beds
14	SLAKT	79	beds
15	BRIMNES	129	beds
16	HEMNES	299	beds
17	ASKVOLL	99	beds
18	MALM	249	beds
19	NEIDEN	39	beds
20	SLATTUM	99	beds
21	RAKKETSTAD	79	wardrobe
22	SONGESAND	149	wardrobe
23	PLATSA	596	wardrobe
24	VUKU	10	wardrobe

25	SMAGORA	99	wardrobe
26	KLEPPSRAD	40	wardrobe
27	KVIKNE	99	wardrobe
28	BRIMNES	99	wardrobe
29	GODISHUS	69	wardrobe
30	PAX	256	wardrobe
31	FABYN	45	chairs
32	LOBERGET	25	chairs
33	LEIFARNE	55	chairs
34	FLINTAN	59	chairs
35	EKEDALEN	45	chairs
36	TEODORES	20	chairs
37	TERJE	13	chairs
38	NILSONE/NORNA	85	chairs
39	JANINGE	29	chairs
40	GUNDE	7	chairs
41	TORARED	20	lamps
42	MUSIK	20	lamps
43	TROSS	8	lamps
44	SJOGANG	9	lamps
45	SINNERLIG	50	lamps
46	LUFTNASSA	13	lamps
47	EBBARED	5	lamps
48	RANARP	20	lamps
49	MELODI	4	lamps
50	BRUNSTA/HEMMA	16	lamps

Виконаємо SQL-запити до таблиці mebel007.orders:

```
-- Структура таблицы `orders`
CREATE TABLE `orders` (
  `OrderID` int(11) NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `ProductID` int(11) NOT NULL,
  `OrderDate` date DEFAULT NULL,
  `OrderQuantity` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `orders`
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (1, 3, 6, '2019-01-29', 2),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (2, 12, 41, '2019-03-06', 1),
```

```

INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (3, 15, 40, '2019-02-01', 3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (4, 9, 5, '2019-02-15', 1),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (5, 10, 3, '2020-02-21', 3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (6, 10, 1, '2020-02-21', 5),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (7, 14, 26, '2019-05-05', 3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (8, 11, 32, '2019-06-26', 7),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (9, 5, 28, '2019-09-30', 2),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (10, 5, 2, '2019-09-30', 3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (11, 7, 49, '2019-12-29', 10),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (12, 13, 24, '2020-02-26', 3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (13, 8, 15, '2020-03-30', 1),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (14, 8, 39, '2020-03-30', 1),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`, `OrderDate`,
`OrderQuantity`) VALUES (15, 2, 10, '2020-04-29', 5);

```

```
-- Индексы таблицы `orders`
```

```

ALTER TABLE `orders`
  ADD PRIMARY KEY (`OrderID`),
  ADD KEY `FK_Orders_Customers` (`CustomerID`),
  ADD KEY `FK_Orders_Products` (`ProductID`);

```

```
-- Ограничения внешнего ключа таблицы `orders`
```

```

ALTER TABLE `orders`
  ADD CONSTRAINT `FK_Orders_Customers` FOREIGN KEY (`CustomerID`) REFERENCES
`customers` (`CustomerID`),
  ADD CONSTRAINT `FK_Orders_Products` FOREIGN KEY (`ProductID`) REFERENCES
`products` (`ProductID`);
COMMIT;

```

Таблиця 6 – Дані таблиці «orders»

OrderID	CustomerID	OrderDate	OrderQuantity	ProductID
1	3	29.01.2019 0:00:00	2	6
2	12	06.03.2019 0:00:00	1	41
3	15	01.02.2019 0:00:00	3	40
4	9	15.02.2019 0:00:00	1	5
5	10	21.02.2020 0:00:00	3	3
6	10	21.02.2020 0:00:00	5	1
7	14	05.05.2019 0:00:00	3	26
8	11	26.06.2019 0:00:00	7	32
9	5	30.09.2019 0:00:00	2	28
10	5	30.09.2019 0:00:00	3	2
11	7	29.12.2019 0:00:00	10	49
12	13	26.02.2020 0:00:00	3	24
13	8	30.03.2020 0:00:00	1	15
14	8	30.03.2020 0:00:00	1	39
15	2	29.04.2020 0:00:00	5	10

ВИСНОВКИ

Figma – це графічний редактор для веб-дизайну, за допомогою нього можна швидко і легко розробляти:

- інтерактивні прототипи сайтів і мобільних додатків;
- спеціальні елементи інтерфейсу – іконки, кнопки, меню, вікна, форми зворотного зв'язку;
- векторні зображення.

Документи програми зберігаються в хмарі, тому одночасно над проектом можуть працювати кілька співробітників, при цьому немає необхідності завантажувати файли для редагування. Вхід в Figma можна здійснити через браузер або ж завантажити програму на свій ПК. Програма працює на Windows і Mac OS. Десктопна версія дозволяє працювати офлайн, а при підключенні інтернету синхронізувати дані для подальшої роботи.

Основні переваги Figma:

- зберігання початкових кодів в хмарі;
- командний доступ до оригіналів макета і можливості для спільної роботи.

Безкоштовний план – це набір базових можливостей без особливих обмежень. Користувачеві доступні функції прототипування, атрибути коду, можливості перегляду і коментування, зберігання файлів без обмежень, імпорт документів з Sketch і ін.

Обмеження в роботі наступні спільне редагування, зберігання версій 30 днів.

Переваги командної роботи доступ до редагування документів; версії зберігаються без тимчасових обмежень; є можливість створення бібліотеки з UI елементами; можна створювати особисті проекти і надавати для них доступи іншим користувачам сервісу.

Дизайнеру важливі швидкість роботи над проектом і комунікація в процесі. Ці та багато інших завдань дозволяє вирішити онлайн-сервіс Figma.

Панель управління програми досить зручна і зрозуміла. Зліва розміщена панель з шарами, включаючи елементи проекту, зверху – панель інструментів, праворуч доступна панель властивостей.

Даний дизайн має свої переваги та недоліки. До переваг дизайну головної сторінки меблевого магазину, можна віднести простоту, достатньо світлий та добре розроблений з точки зору UI та UX-принципів. До недоліків даного сайту, потрібно віднести негармонічне поєднання кольорів, що в деяких моментах погано читається користувачем також погано продуманий блок «Про “МЕВЕЛ 007”».

Створений сайт відповідає всім вимогам, що пред'являються до подібного роду програмного забезпечення. В наданій розробці реалізовані всі важливі функції необхідні для роботи.

У даній роботі були розглянуті наступні питання:

- проаналізовано меблевий ринок;
- створення дизайну сайту в приємному колірному виконання;
- розробка презентаційного сайту з використанням мови гіпертекстової розмітки HTML;
- створення графічних елементів сайту;
- верстка дизайну сайту на шаблон;

Сайт дає можливість зв'язатися по електронній пошті з адміністрацією підприємства для з'ясування і уточнення будь-яких питань, а також для замовлення продукції.

Безумовно, створений красивий і приємний зовнішній вигляд сайту. У міру розвитку підприємства і розширення асортименту, що випускається, інформація на сайті може бути змінена і доповнена, так як даний програмний продукт досить гнучкий і масштабується. Таким чином, мета повністю реалізована. Поставлені завдання виконані. Створений сайт компанія використовує в мережі в рекламних цілях. Незаперечним плюсом є те, що не

потрібно установка програмного продукту, а наявність бази даних забезпечить можливість своєчасного оновлення прайс-листів, дозволить користувачам отримувати актуальну інформацію про надходження нових товарів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Что такое ON-LINE сервисы. URL: <https://avada-media.ua/services/on-line-servisy/> (дата звернення: 21.03.2020).
2. Что такое Figma: возможности и принципы работы. URL: https://skillbox.ru/media/design/chto_takoe_figma/ (дата звернення: 23.03.2020).
3. Обзор программы Figma. URL: <http://figmadesign.ru/1-0-0-obzor-figma.html> (дата звернення: 29.03.2020).
4. Чем Figma лучше Photoshop для разработки веб-интерфейсов. URL: <https://habr.com/ru/company/skillbox/blog/359258/> (дата звернення: 12.04.2020).
5. Возможности Figma. Сервис для разработки веб-дизайна. URL: <https://hyperhost.ua/info/vozmozhnosti-figma-obzor-instrumenta-dlya-sozdaniya-veb-dizauna/> (дата звернення: 19.04.2020).
6. Основные понятия БД. URL: <http://informatic.ugatu.ac.ru/lib/office/Proekt.htm> (дата звернення: 21.04.2020).
7. 10 лучших инструментов для разработки и администрирования MySQL. URL: <https://habr.com/ru/post/142385/> (дата звернення: 01.05.2020).

ДОДАТКИ

ДОДАТОК А

Виконання SQL-запитів до таблиці mebel007.customers

```

-- Структура таблиці `customers`
CREATE TABLE `customers` (
  `CustomerID` int(20) NOT NULL,
  `FirstName` text,
  `SecondName` text,
  `Phone` varchar(24) DEFAULT NULL,
  `Email` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп даних таблиці `customers`
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (1, 'Alexandr', 'Baver',
'0670423123', 'baver63@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (2, 'Simona', 'Vaychute',
'0639568963', 'simona_vaychute@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (3, 'Viktoriiia',
'Bodnar', '0950263741', 'bodnar-vik@mail.ru'),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (4, 'Anastasia',
'Golovey', '0676793672', 'golovey20@ukr.net'),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (5, 'Roman',
'Lykiyanenko', '(095)6767321', NULL),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (6, 'Igor', 'Babayev',
'0973753147', 'babayev96@gmail.com'),
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (7, 'Victoria',
'Matunyk', '0632930532', 'vic.martunyk@gmail.com'),

```

```
INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (8, 'Dasha', 'Kozuba',
'0666578357', 'kozuba2017@ukr.net'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (9, 'vitaliy', 'Veslov',
'0935388321', 'veslovvvv1@mail.com'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (10, 'Vladislav',
'Fomin', '0938787222', 'vlad.fomin@gmail.com'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (11, 'Masha', 'Lozuk',
'0639393987', 'lozuk_masha@ukr.net'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (12, 'Ekaterina',
'Vistoma', '0503232369', 'vistoma_1993@ukr.net'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (13, 'Egor', 'Krid',
'0987417411', 'krid_krid123@mail.com'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (14, 'Roma', 'Gorban',
'0639551755', 'r_gorban@mail.com'),

INSERT INTO `customers` (`CustomerID`, `FirstName`,
`SecondName`, `Phone`, `Email`) VALUES (15, 'Karina', 'Mileeva',
'0673941555', 'mileeve.karina@gmail.com');
```

```
-- Индексы таблицы `customers`
```

```
ALTER TABLE `customers`
  ADD PRIMARY KEY (`CustomerID`);

COMMIT;
```

ДОДАТОК Б

Виконання SQL-запитів до таблиці mebel007.products

```

-- Структура таблицы `products`
CREATE TABLE `products` (
  `ProductID` int(20) NOT NULL,
  `ProductName` text,
  `Price` int(11) DEFAULT NULL,
  `TypeOfFurniture` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `products`
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (1, 'ЕКТОРР', 349, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (2, 'KIVIK', 69, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (3, 'BARNARP', 149, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (4, 'FRIHETEN', 339, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (5, 'BRUNSTA/HEMMA', 16, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (6, 'KNOPPARP', 79, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (7, 'RAGUNDA', 449, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (8, 'SLATORP', 799, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (9, 'ASARUM', 199, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (10, 'VIMLE', 849, 'sofas'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (11, 'NEIDEN', 75, 'beds'),

```

```

INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (12, 'SONGESAND', 261, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (13, 'DUNVIK', 824, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (14, 'SLAKT', 79, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (15, 'BRIMNES', 129, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (16, 'HEMNES', 299, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (17, 'ASKVOLL', 99, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (18, 'MALM', 249, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (19, 'NEIDEN', 39, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (20, 'SLATTUM', 99, 'beds'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (21, 'RAKKETSTAD', 79, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (22, 'SONGESAND', 149, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (23, 'PLATSA', 596, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (24, 'VUKU', 10, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (25, 'SMAGORA', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (26, 'KLEPPSRAD', 40, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (27, 'KVIKNE', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (28, 'BRIMNES', 99, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (29, 'GODISHUS', 69, 'wardrobe'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (30, 'PAX', 256, 'wardrobe'),

```



```

INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (31, 'FABYN', 45, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (32, 'LOBERGET', 25, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (33, 'LEIFARNE', 55, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (34, 'FLINTAN', 59, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (35, 'EKEDALEN', 45, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (36, 'TEODORES', 20, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (37, 'TERJE', 13, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (38, 'NILSONE/NORNA', 85, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (39, 'JANINGE', 29, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (40, 'GUNDE', 7, 'chairs'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (41, 'TORARED', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (42, 'MUSIK', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (43, 'TROSS', 8, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (44, 'SJOGANG', 9, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (45, 'SINNERLIG', 50, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (46, 'LUFTNASSA', 13, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (47, 'EBBARED', 5, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (48, 'RANARP', 20, 'lamps'),
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,
`TypeOfFurniture`) VALUES (49, 'MELODI', 4, 'lamps'),

```

```
INSERT INTO `products` (`ProductID`, `ProductName`, `Price`,  
`TypeOfFurniture`) VALUES (50, 'KNISILINGE', 279, 'sofas');
```

```
-- Индексы таблицы `products`
```

```
ALTER TABLE `products`
```

```
ADD PRIMARY KEY (`ProductID`);
```

```
COMMIT;
```

ДОДАТОК В

Виконання SQL-запитів до таблиці mebel007.orders

```
-- Структура таблицы `orders`  
CREATE TABLE `orders` (  
  `OrderID` int(11) NOT NULL,  
  `CustomerID` int(11) NOT NULL,  
  `ProductID` int(11) NOT NULL,  
  `OrderDate` date DEFAULT NULL,  
  `OrderQuantity` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-- Дамп данных таблицы `orders`  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (1, 3, 6, '2019-01-29', 2),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (2, 12, 41, '2019-03-06',  
1),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (3, 15, 40, '2019-02-01',  
3),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (4, 9, 5, '2019-02-15', 1),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (5, 10, 3, '2020-02-21',  
3),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (6, 10, 1, '2020-02-21',  
5),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (7, 14, 26, '2019-05-05',  
3),  
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,  
  `OrderDate`, `OrderQuantity`) VALUES (8, 11, 32, '2019-06-26',  
7),
```

```

INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (9, 5, 28, '2019-09-30',
2),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (10, 5, 2, '2019-09-30',
3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (11, 7, 49, '2019-12-29',
10),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (12, 13, 24, '2020-02-26',
3),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (13, 8, 15, '2020-03-30',
1),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (14, 8, 39, '2020-03-30',
1),
INSERT INTO `orders` (`OrderID`, `CustomerID`, `ProductID`,
`OrderDate`, `OrderQuantity`) VALUES (15, 2, 10, '2020-04-29',
5);

```

```
-- Индексы таблицы `orders`
```

```

ALTER TABLE `orders`
  ADD PRIMARY KEY (`OrderID`),
  ADD KEY `FK_Orders_Customers` (`CustomerID`),
  ADD KEY `FK_Orders_Products` (`ProductID`);

```

```
-- Ограничения внешнего ключа таблицы `orders`
```

```

ALTER TABLE `orders`
  ADD CONSTRAINT `FK_Orders_Customers` FOREIGN KEY
(`CustomerID`) REFERENCES `customers` (`CustomerID`),
  ADD CONSTRAINT `FK_Orders_Products` FOREIGN KEY (`ProductID`)
REFERENCES `products` (`ProductID`);
COMMIT;

```