

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток конструювання верхнього одягу»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектуювання»

Виконавець роботи: студентка групи ІТ-62 Ковенко Анна Станіславівна

Кваліфікаційна робота бакалавра

захищена на засіданні ЕК

з оцінкою

_____ «__» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Кузнецов Е. Г.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2020 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Ковенко Анна Станіславівна

1 Тема роботи *Web-додаток конструювання одягу*

керівник роботи *_____ Кузнєцов Е.Г., к.т.н., доцент _____,*

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи: зображення частин одягу, інформація про оціни

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- 1) Аналіз предметної області
- 2) Моделювання та проектування
- 3) Практична реалізація

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): постановка задачі, дослідження аналогів, вимоги до роботи, порівняльна характеристика аналогів, вхідні дані, контекстна діаграма процесу розробки, діаграма декомпозиції процесу розробки, діаграма варіантів використання, вибір мови програмування, розробки прототипу, розробка візуальної частини, розробки функціональної частини, публікація сайту в мережі інтернет.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2019

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	До 08.03.2020	
2	Оформлення технічного завдання	До 03.04.2020	
3	Проведення аналізу предметної області	До 08.04.2020	
4	Проведення структурно-функціонального моделювання процесів	До 05.05.2020	
5	Розробка проекту	До 10 05.2020	
6	Публікація в мережі інтернет	До 19 05.2020	
7	Здача пояснювальної записки та файлів розробленого проекту	До 01.06.2020	

Студент _____
(підпис)

Ковенко А.С.

Керівник роботи _____
(підпис)

к.т.н., доц. Кузнецов Е.Г

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток конструювання одягу». Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 21 найменувань, додатків. Загальний обсяг роботи – 68 сторінок, у тому числі 38 сторінок основного тексту, 2 сторінки списку використаних джерел, 27 сторінок додатків.

Метою кваліфікаційної роботи бакалавра є розробка оригінального та функціонального конструктору одягу з функціями інтернет магазину, який надаватиме можливість користувачам створювати власний дизайн одягу і замовляти не виходячи з дому.

В роботі проведено аналіз аналогів, проектування і розробку додатку. Результатом проведеної роботи є Web-додаток конструювання одягу. Практичне значення роботи полягає у застосуванні сучасних технологій, які допоможуть користувачеві розробити власний дизайн одягу.

Ключові слова: PHP, JAVASCRIPT, ОДЯГ, ПРОТОТИП, СКРИПТИ, JQUERY, САЙТ

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дослідження предметної області.....	7
1.2 Аналіз програмних продуктів – аналогів	10
1.3 Вибір засобів реалізації	13
1.4 Постановка задачі	18
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	20
2.1 Моделювання процесу роботи сайту	20
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	26
3.1 Архітектура веб додатку	26
3.2 Програмна реалізація.....	26
3.3 Використання програмного додатку.....	31
ВИСНОВОК.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ.....	40
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ.....	45
ДОДАТОК В. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	58

ВСТУП

Розвиток Інтернет-технологій завжди був рушійною силою для виникнення нових видів сайтів. Почали з'являтися блоги, портали, форуми, інтернет магазини і у людей з'явилась можливість спілкуватися, отримувати відповіді на будь-які питання і навіть проводити ділові операції. В наш час стає все більш поширеним придбання будь-яких товарів в інтернет-магазинах.

На сьогоднішній день все більше людей почали усвідомлювати переваги Інтернет-технологій. Однією з таких переваг є покупки в Інтернеті. Скориставшись послугами віртуальної торгівлі, можна спокійно і обдуманно зробити необхідну покупку в будь-який час доби. Сьогодні інтернет-магазини та веб-сервіси беруть активну участь у нашому житті, і саме зараз в наше життя дуже стрімко входять онлайн інтернет магазини, веб-сервіси, тому все частіше ми здійснюємо покупки онлайн, завдяки лише декількох кліків миші.

Метою кваліфікаційної роботи бакалавра є розробка оригінального та функціонального конструктору одягу з функціями інтернет магазину, який надаватиме можливість користувачам створювати власний дизайн одягу і замовляти не виходячи з дому.

Для реалізації поставленої мети потрібно вирішити такі задачі:

- Вивчити процес конструювання і замовлення одягу.
- Провести аналіз сайтів-аналогів для розроблення технічного завдання.
- Обрати та налаштувати інструменти реалізації.
- Визначити структуру даних та спроектувати веб-сайт.
- Реалізувати додатки та розробити інструкції користувача.

Результатом проекту буде веб-додаток, що надаватиме змогу потенційним клієнтам мати повне уявлення про речі, які вони збираються замовити за власним сконструйованим зразком.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження предметної області

Сайт дозволяє структурувати інформацію, представлену у всесвітній павутині завдяки використанню сторінок. Наприклад, сайт ІВМ містить тисячі сторінок. Перша сторінка сайту (головна сторінка) завдяки використанню меню забезпечує навігацію по усім іншим сторінкам.

Сторінка веб-сайту – це набір текстових файлів з тегами HTML. Коли відвідувач завантажує ці файли на свій комп'ютер, вони обробляються браузером і виводяться на інструмент відображення користувача. HTML дозволяє формувати текст, виділяти функціональні елементи, створювати гіпертекстові посилання та вставляти на сторінку зображення, записи та інші мультимедійні елементи. Можливо змінити відображення сторінки, додавши стиль CSS, який дозволяє групувати всі елементи форматування (розмір та колір великих літер, розмір та тип вставлених блоків тощо) або сценарій на мові JavaScript, за допомогою якого є можливість переглядати сторінки з подіями.

Сайти можуть містити підрозділи, орієнтовані цілком на ту чи іншу аудиторію. Кожен сайт створюється з якоюсь певною метою. Перш за все, вони потрібні для передачі певної інформації користувачу мережі.

Відповідно до інформаційного змісту веб-сайту, сайти можна розділити на сайти-візитки, що містять загальну інформацію власників веб-сайтів чи організацій; представницькі сайти, які іноді також називаються сайтами візитних карток із розширеними функціями, такими як: послуги, портфоліо, зворотній зв'язок, форма зворотного зв'язку, детальний опис; сайт компанії, який містить повну інформацію про компанію власника, послуги, продукцію, події в житті компанії. Він відрізняється від сайтів візитних карток та представницьких сайтів цілісністю наданої інформації, і зазвичай містить різні функціональні інструменти для обробки вмісту (пошук та фільтри, календарі

подій, фотогалереї, блоги компаній, форуми). Однак найпопулярнішим типом веб-сайту є інтернет-блог. Блог – це інтернет-журнал, основним наповненням якого є записи, що постійно публікуються. Записи включають текст, фотографії, графіку або мультимедіа. Записи в блозі зазвичай короткі та розташовані у зворотному хронологічному порядку. У більшості випадків читачі публічних блогів вступають в полеміку з авторами блогу через публікації коментарів чи особисті блоги. Види блогів наведені в таблиці 1.1.

Таблиця 1.1 – Типи веб-сайтів

Назва	Опис
Блоги про подорожі	Блоги для подорожей мають величезний потенціал для залучення багато трафіку та заробітку. Люди люблять подорожувати, і вони завжди шукатимуть нових можливостей, щоб поїхати кудись, де раніше не були, і вивчити нові культури.
Блоги про здоров'я	Ця ніша також має великі партнерські можливості, оскільки можна писати про продукти харчування, добавки, продукти для фізичних вправ та багато іншого.
Блоги електронного навчання	Електронне навчання зростає блискавично, оскільки інтернет зробив навчання та знання доступними для всіх. Завжди знайдуться люди, які прагнуть отримати нові знання та навички в інтернеті, не виходячи з дому. Можна давати уроки гітари, уроки кодування, уроки щодо використання будь-якого програмного забезпечення та багато іншого.
Технологічні та ігрові блоги	Можна писати огляди додатків та відеоігор, огляди технічних гаджетів, поради щодо ремонту мобільних пристроїв та ПК, навчальні відео-ігри чи будь-яку іншу тему.

Продовження таблиці 1.1

Назва	Опис
Блоги для батьків	Є кілька тем, що стосуються догляду за дітьми та надання допомоги новим мамам вибирати найкращу їжу, одяг, безпечні іграшки та розважальні заходи для своїх дітей або планувати день народження.
Блоги спільноти	Блоги спільноти створені для того, щоб об'єднати однодумців. Ці блоги зазвичай висвітлюють теми, що цікавлять однодумців, і містять розділ коментарів та сильну підтримку соціальних медіа для сприяння спілкуванню.
Живі блоги	<p>Писати про світові напрямки у своєму блозі – це одне, але показувати їх у реальному часі своїй аудиторії – це інше.</p> <p>Блоги веб-камер, що живуть, включають також ті, де творці вмісту не пишуть вміст, а вербально передають його аудиторії. У поєднанні з чатом у прямому ефірі цей вид ведення блогів є дуже привабливим, а також може бути прибутковим.</p>
Суміші всіх типів	І нарешті, у є блоги, які виступають як суміші всіх типів, про які було сказано раніше. Ці блоги можуть висвітлювати різні теми, націлювати на кілька демографічних даних та містити всі типи мультимедійних форматів.

Сайти для конструювання одягу являються досить зручними для користувача, а також вигідними для власників інтернет магазинів. Такі сайти в наш час набирають дуже велику популярність з боку, як користувачів, так і

з боку підприємців. Це веб-сайти, в яких кожна людина може купити будь-який товар не відходячи з дому.

Продукт проекту може використовувати не тільки замовник (наприклад, Петро Сорока власник магазинів одягу в Сумах), але і іншими власниками магазинів для просування власної марки одягу та послуг.

Такі сайти в даний час дуже популярні, адже вони надають можливість користувачам створювати речі за власними моделями, такими як пальто та сукні, тому при виготовленні в реальному світі не буде проблем при виборі одягу.

Створення такого додатку є актуальним питанням, оскільки існуючі аналоги зазвичай являються платними, а також з обмеженими можливостями. На відміну від вже існуючих додатків даний продукт є доступним та враховує основні недоліки попередників.

Таким чином, стильний і сучасний сайт формує уявлення про модний магазин одягу, представляє його у вигідному ракурсі перед потенційними клієнтами і при цьому не вимагає значних зусиль і матеріальних витрат для підтримки функціонування.

1.2 Аналіз програмних продуктів – аналогів

Процес розробки сайту конструювання одягу неможливий без попередньої постановки завдань і цілей а також ретельного аналізу ринкової ніші, у якій працює компанія. Аналіз веб-сайтів конкурентів, аналіз цільової аудиторії майбутнього сайту, виявлення «сильних» і «слабких» сторін проекту. Розробка комерційного сайту завжди починається з виконання подібних завдань.

Для досягнення мети було проведено дослідження сторінки веб-сайтів:

- tailorgeorge.com.
- just-tailor.com.

Сайт tailorgeorge призначений для продажу чоловічих сорочок, має досить зручний і сучасний інтерфейс, зручний і зрозумілий конструктор для створення власної сорочки.

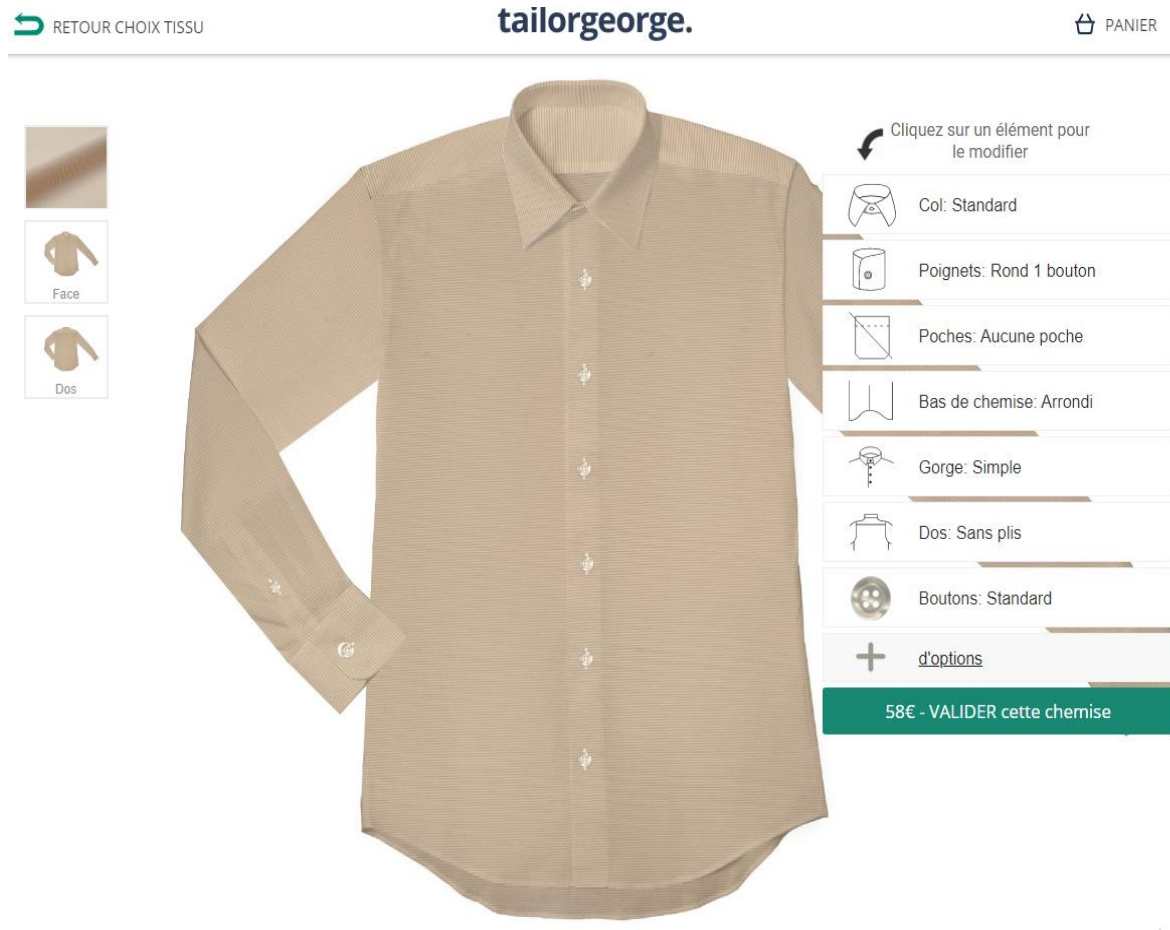


Рисунок 1.1 – Головна сторінка сайту конструювання сорочки

Проте даний сайт має ряд недоліків:

- Відсутність різноманіття, можливо конструювати лише сорочку.
- Недопрацьований контент, іноді відсутні фото.
- Відсутність можливості замовлення без реєстрації чи авторизації.

До переваг сайту також можна віднести:

- Зручна панель навігації.
- Можливість вибору тканини та інших деталей.
- присутність цінової політики.

Ще один сайт зі схожою тематикою і призначення це just-tailor.

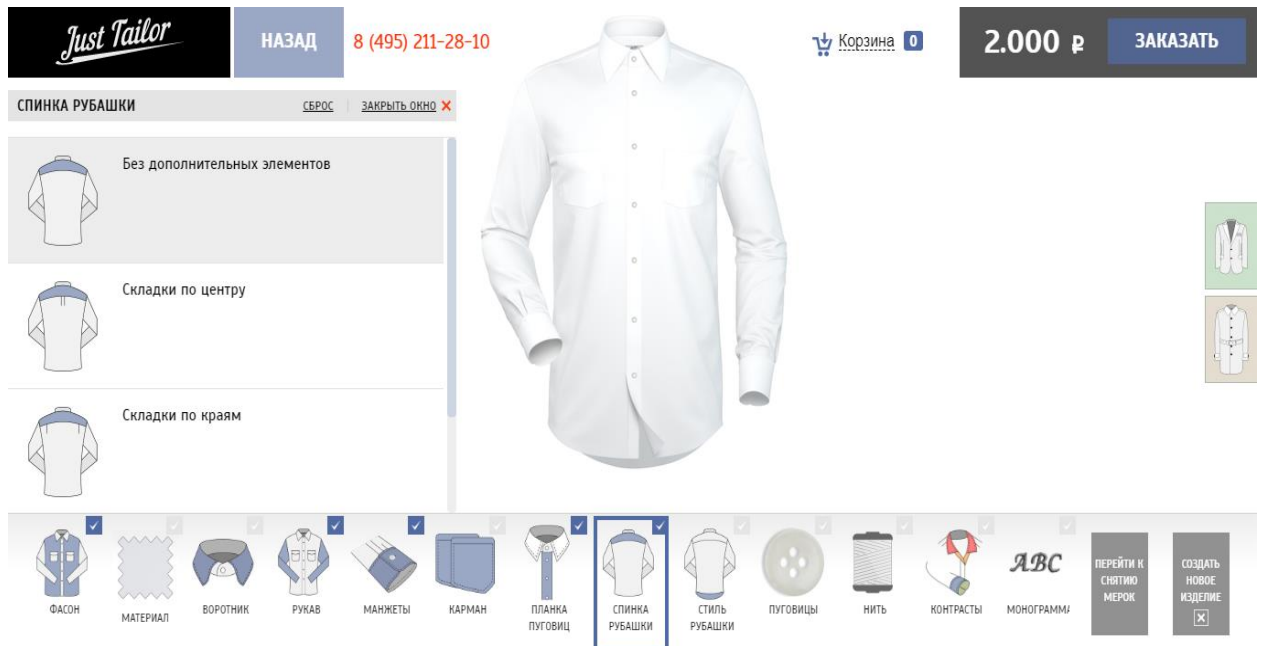


Рисунок 1.2 – Головна сторінка сайту just-tailor

Даний сайт також призначений для продажу одягу і має конструктор для створення власного дизайну одягу. Проте як і попередній сайт він також має свої переваги і недоліки.

Даний сайт має ряд недоліків:

- Не зручна панель навігації.
- Елементи контенту мають не досить компактне розміщення на сайті.

До переваг сайту також можна віднести:

- Можливість вибору тканини та інших деталей.
- Присутність цінової політики.

Проаналізувавши веб-сторінки конструювання одягу в пошуковій системі, які представляються користувачеві було проведено порівняльний аналіз (табл.1.1), де було виявлено позитивні та негативні сторони, які при створенні дипломного проекту треба уникати.

Таблиця 1.1 – Аналіз розглянутих сайтів конструювання одягу

Критерії	Web-сайт tailorgeorge	Web-сайт just-tailor	Веб-додаток Модний бутик(мій)
Відсутність непотрібної реклами	+	+	+
Різноманіття одягу	-	+	+
Можливість зворотного зв'язку	-	+	+
Компактність елементів контексту	-	-	+
Зручна панель навігації	+	-	+
можливість вибору тканини та інших деталей	+	+	+
Можливість замовлення без реєстрації чи авторизації	-	+	+
Видима цінова політика	+	+	+

Отже, в результаті проведеного аналізу сайтів зі схожою тематикою і призначенням було вирішено розробити сайт який не матиме недоліків виявлених на сайтах представлених в таблиці 1.1.

Дана розробка матиме всі переваги даних сайтів, а також матиме унікальний дизайн та зручний інтерфейс.

1.3 Вибір засобів реалізації

Для реалізації веб-ресурсу було обрано такі інструменти реалізації: мову програмування PHP загального призначення та мову програмування JavaScript, з використанням бібліотеку jQuery.

PHP – це сценарна мова загального призначення, яка широко використовується в розробці веб-додатків. В даний час вона взаємодіє з великою кількістю постачальників послуг хостингу і є однією з лідерів мов програмування, що використовуються при розробці веб-сайтів. Ця мова програмування спеціально розроблена для розробки веб-сторінок і може бути реалізована безпосередньо в коді веб-сторінки [1].

Мова програмування PHP включає безліч вбудованих функцій, таких як: функції файлової системи, функції протоколу HTTP, функції дати та часу, обробка рядків та масивів тощо.

Основна особливість мови полягає в тому, що вона може добре взаємодіяти з усіма сучасними веб-технологіями та підтримує велику кількість сучасних веб-протоколів, а саме: прикладного рівня для доступу до IMAP електронної пошти, FTP, POP, SNMP тощо. PHP також дуже підходить для використання баз даних. Більшість сучасних СУБД підтримують мову скриптів PHP.

Особливості мови PHP включають багато надійних механізмів шифрування. Крім того, сценарії, написані на PHP, сумісні з багатьма сторонніми програмами. Це дозволяє легко інтегруватися із захищеною технологією електронної комерції.

Ще однією перевагою цієї мови є те, що вихідний текст скрипту PHP не можна переглянути в браузері, оскільки він складається безпосередньо перед надсиланням за бажанням користувача. У результаті сценарій не дозволяє користувачам красти вихідний код оригінальних команд PHP [2].

JavaScript – мова програмування для управління сценаріями перегляду веб-сторінок. Основна особливість цієї мови полягає в тому, що при використанні цієї мови елементи середовища відображення можуть бути змінені під час перегляду веб-сайту, тому веб-сторінку не буде

перезавантажено. Наприклад, за допомогою JavaScript можна змінити колір фону веб-сторінки, замінити зображення, інтегровані у веб-сторінку, створити нове вікно відображення, або відобразити повідомлення [3].

JavaScript є об'єктно-орієнтованою мовою програмування, але оскільки вона використовує прототипи (швидко реалізують основні функції для аналізу системи), порівняно з традиційними об'єктно-орієнтованими, існує багато функціональних мов програмування для прямого виконання мовного коду. Крім того, JavaScript має перелік атрибутів, притаманних іншим функціональним мовам, а саме: функції як об'єкти, об'єкти як списки та анонімні функції.

Хоча мова JavaScript по синтаксису схожа на мову C, вона все ж має ряд відмінностей:

- Використовуються об'єкти зі здатністю самоаналізу.
- Функції як об'єкти класу.
- Автоматичне приведення типів.
- Анонімні функції.

Як недоліки мови JavaScript можна виділити наступні:

- Немає можливості регулювати області.
- Відсутність інтерфейсу як такого.
- використання стандартних інтерфейсів доступу до Web-серверів і баз даних [4].

jQuery – бібліотека на базі JavaScript, призначена для керування веб-сторінками, що функціонують на різних платформах. Крім того, jQuery включає інструменти, які допомагають фіксувати та обробляти події, створювати анімацію на веб-сторінках та методи, що дозволяють взаємодіяти з сервером без перезавантаження сторінки (AJAX).

Основна причина, чому бібліотека jQuery популярна, полягає в тому, що при виконанні типових завдань не потрібно створювати великі конструкції JavaScript, які можуть бути доступні не у всіх браузерах. Це пояснюється тим,

що один браузер може підтримувати певні об'єкти, властивості або методи JavaScript, а інший таких можливостей не має. У цьому випадку для браузера повинен бути записаний якийсь інший код, щоб браузер підтримував операції, запрограмовані в коді. А оскільки браузерів багато, знадобиться багато роботи, і лише написані сценарії на jQuery можуть працювати однаково у всіх браузерах.

Переваги jQuery:

- Кросбраузерність. Синтаксис бібліотеки підтримується всіма сучасними браузерами.
- Компактний код. Робота, яку JavaScript повинен виконувати в окремій функції, реалізована в кількох рядках коду через jQuery. І такий код стислий, логічний і простий для розуміння.
- Зручно обробляти події та візуальні ефекти. Такі завдання, як створення підказки на миші, запуск анімації при натисканні і запрошення користувачів даних легко реалізуються на jQuery. JavaScript також може вирішити цю проблему, але в цьому випадку код є більш громіздким і менш гнучким.
- Зрозуміла документація. На офіційному ресурсі можна ознайомитися з усіма її функціями бібліотеки [5].

Для зберігання даних часто використовують JSON. JSON або JavaScript Object Notation – це формат, який може реалізувати неструктуроване подання тексту структурованих даних на основі принципів пар ключових значень та упорядкованих списків. Незважаючи на те, що JSON поширюється з JavaScript, більшість мов спочатку підтримували JSON через спеціальні бібліотеки. Зазвичай Json використовується для обміну інформацією між веб-клієнтом та веб-сервером.

JSON – це загальний формат даних з мінімальною кількістю типів значень: рядок, число, булевий (один або нуль), список, об'єкт і нуль. Хоча JSON – це підмножина JavaScript, цей тип даних існує в більшості популярних

мов програмування, що робить JSON ідеальним вибором для передачі даних між програмами, написаними різними мовами [6].

За останні 15 років JSON став офіційним стандартом обміну даними та застосовується майже будь-де в Інтернеті. Сьогодні майже всі веб-сервери цим користуються. Ця популярність пояснюється також тим, що багато баз даних підтримують JSON.

Дві найпопулярніші мови розмітки – HTML та XML. HTML розшифровується як мова розмітки HyperText та використовується для створення веб-сайтів. Додані до звичайного текстового документа, всі теги HTML описують, як цей документ повинен відобразитися веб-браузером [7].

Невідмінна частина будь-якої веб сторінки – каскадні стилі розмітки сторінки або CSS. Таблиці стилів CSS – це формальна мова, яка використовується для опису зовнішнього вигляду документів, написаних мовою розмітки. Розробники веб-сайтів використовують CSS для встановлення шрифтів, їх макета на сторінці та інших принципів того, як виглядають блоки веб-сторінок. Основним завданням створення CSS є стилістичне відображення контенту, тим самим збільшуючи доступність документа та забезпечує більшу гнучкість та можливість керування відображенням веб-сторінок та зменшення складності їх структури [8].

Хоча HTML використовується для структури веб-документа, визначає такі речі, як заголовки та абзаци, і дозволяє вставляти зображення, відео та інші медіа, використаний CSS визначає стиль створеного документа.

Для розробки клієнтської частини сайт було обрано мови програмування HTML, CSS і JavaScript так як вони є най популярнішими мовами для веб розробки і досить легкі у вивченні. Для зберігання даних буде використовуватись JSON.

Розробка серверної частини буде відбуватись за допомогою мови програмування PHP. Ця мова активно використовується багатьма сайтами і досить легка у вивченні.

1.4 Постановка задачі

Виходячи із визначеної вище актуальності проблеми, було визначено, що потрібно створити веб-додаток для того, щоб кожен бажаючий міг розробити власний дизайн одягу і замовити його не виходячи з дому. Основною аудиторією сайту будуть користувачі з базовими навичками користування комп'ютером.

Проект повинен бути реалізований у вигляді веб-додатку, доступного в мережі Інтернет. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

Даний додаток повинен мати наступні можливості:

- Огляд одягу з різних боків при натисканні на відповідні кнопки або стрілочки.
- Зміна, видалення або додавання нових елементів одягу.
- Можливість конструювання не лише верхнього одягу, а і суконь.
- Крім можливості вибору дизайну, повинна бути додана функція яка зможе одразу показувати ціну кожного елемента, а також функція яка сумує усі елементи та буде показувати кінцеву ціну, ще до того як перейти до замовлення.

Також додаток повинен відповідати реальному вигляду окремих деталей речей, всім вимогам отриманим у технічному завданні, а також має бути утверджений замовником.

Веб-додаток, повинен володіти наступними особливостями:

- Сайт повинен бути повністю функціональний, а також мати інтуїтивно зрозумілий інтерфейс.

– Веб-сервіс повинен підтримувати графічні вставки, та анімації, які є дуже важливими для процесу конструювання одягу, та відображення цілісної картинки.

– Повинна бути також реалізована можливість замовлення онлайн, після чого менеджер зворотнім зв'язком з користувачем з'ясує деталі замовлення.

Основним завданням проектування є створення такої системи конструювання одягу, яка б дозволяла за різними частинами деталей одягу сконструювати власну річ, а також вносити зміни у адміністративній панелі для людей, які не мають навичок у створенні веб-сайтів.

Для реалізації поставленої мети потрібно вирішити такі задачі:

1. Постановка цілей і завдань сайту.
2. Створення та опрацювання технічного завдання (ТЗ) на розробку сайту.
3. Вивчити процеси організації конструювання та замовлення одягу.
4. Обрати та налаштувати інструменти реалізації.
5. Проектування та розробка сайту.
6. Наповнення контентом.
7. Тестування сайту та розміщення в Інтернет мережі.

Детальна інформація щодо розробки веб-додатку наведена у розробленому в ході аналізу предметної області технічному завданні (Додаток А).

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Моделювання процесу роботи сайту

2.1.1 Моделювання процесу роботи спортивної секції у нотації IDEF

Функціональна модель IDEF0 – це набір блоків, кожен блок – це "чорна скринька", а його вхідні та вихідні, елементи управління та механізми були детально розкладені до необхідного рівня. За допомогою стрілок та описів функціонального блоку функції пов'язані між собою. Крім того, кожен тип стрілки чи дії має своє значення. Ця модель дозволяє описати всі основні типи процесів, включаючи управління та організаційні процеси [14].

На нульовому рівні весь процес розглядається як один функціональний блок з усіма пов'язаними робочими і керуючими об'єктами. На цій діаграмі також відбивається мета структурного аналізу і точка зору, з позиції якої розглядається модель. Діаграма нульового рівня даного проекту відображена на рисунку 2.1.

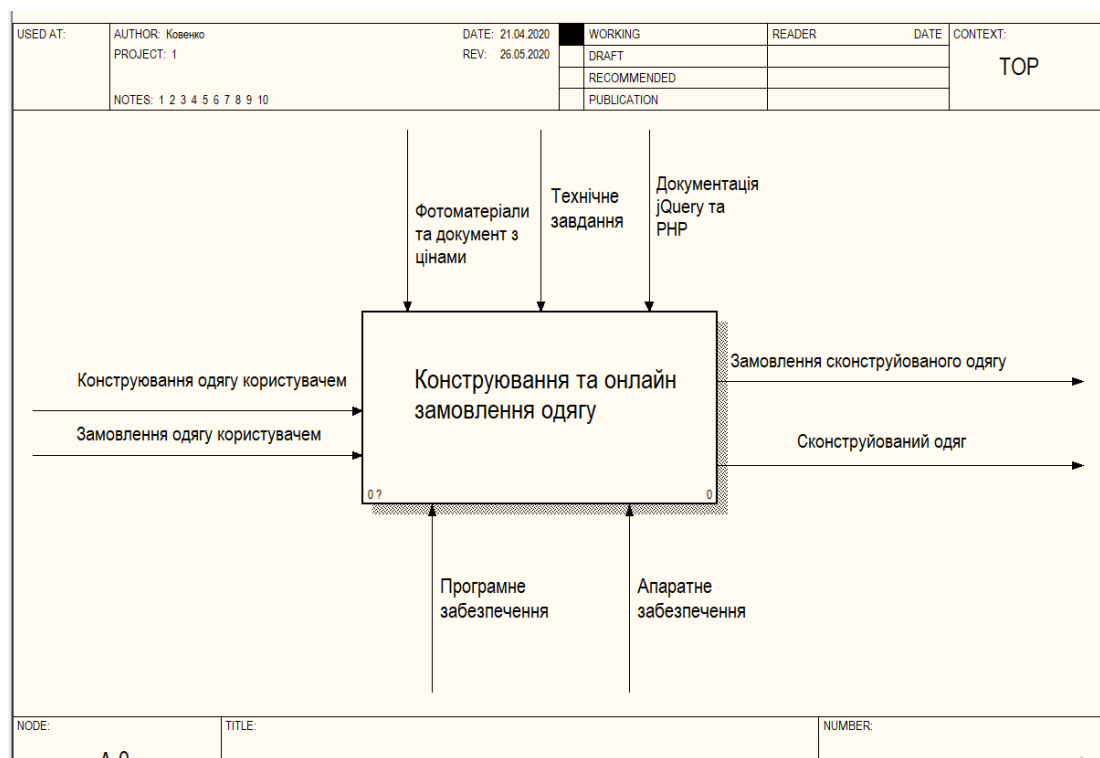


Рисунок 2.1 – Діаграма IDEF 0

Перша діаграма рівнів детально описує функцію обробки нульового рівня. Так, функціональний блок 0 розкладається на сукупність взаємопов'язаних підфункцій. Представлений варіант діаграми для даного проекту представлений на рисунку 2.2.

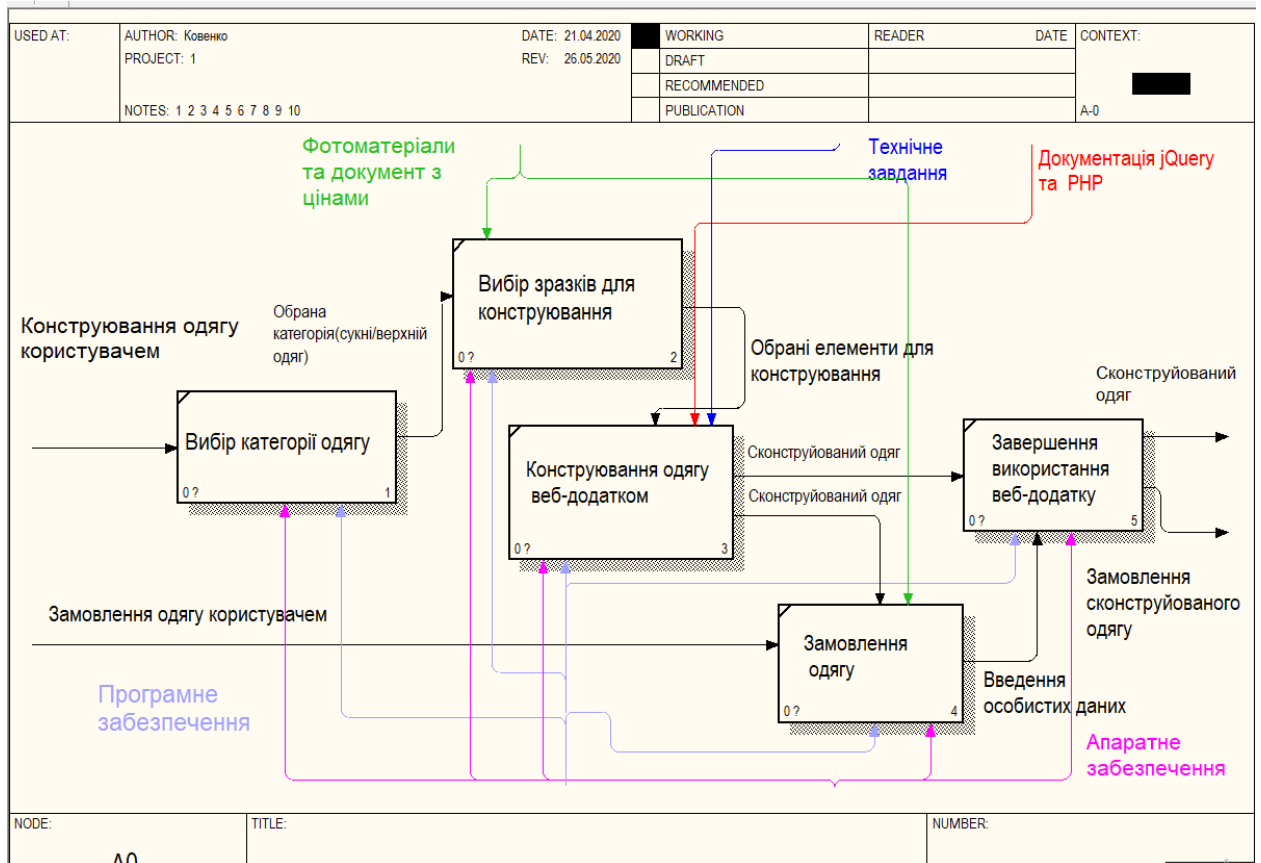


Рисунок 2.2 – Діаграма IDEF 1

2.1.2 Модель варіантів використання веб-додатку

Діаграма варіантів використання є найбільш загальним поданням функціональних вимог до системи. Для подальшого проектування системи потрібні більш конкретні деталі, які описуються в документі, який називається сценарієм варіантів використання або потоком подій.

Сценарій детально документує процес взаємодії дійової особи з системою, що реалізується в рамках варіанту використання. Основний потік подій описує нормальний хід подій. Альтернативні потоки описують

відхилення від нормального перебігу подій і їх обробку. Основні елементи діаграми – учасник і прецедент.

Учасник – це безліч логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами або сутностями. Учасником може бути людина або інша система, підсистема або клас. Графічно учасник зображується "чоловічком".

Прецедент – опис безлічі послідовних подій, що виконуються системою, які призводять до спостережуваного учасником результату. Прецедент не показує, "як" досягається певний результат, а тільки "що" саме виконується. Прецеденти позначаються дуже простим чином – у вигляді еліпса, всередині якого вказано його назву [15].

На малюнку 2.3 відображена модель варіантів використання для веб додатку конструювання одягу. Інформація про акторів та опис варіантів використання описана в таблицях 2.1 та 2.2.

Табл. 2.1 – Опис акторів

Назва	Опис
Актори-користувачі	
Користувач	Користувач веб додатку конструктору одягу.
Адміністратор	Користувач, що має повний доступ до можливостей додатку.

Табл. 2.2 – Опис варіантів використання

Назва	Опис
Оформлення замовлень	Створення заявки замовлення обраного товару на сайті.
Підбір фасону	Можливість обрати фасон одягу відповідно вимогам користувача.

Вибір ціни	Можливість обрати ціну одягу відповідно вимогам користувача.
Редагування даних	Адміністратор може редагувати дані на сайті.
Перегляд замовлень	Адміністратор може переглядати замовлення користувачів сайту.



Рисунок 2.3 – Діаграма варіантів використання

2.1.3 Модель аналізу розроблюваного ПЗ

За допомогу діаграми прецедентів відбувається представлення багатьох об'єктів, які мають однакову структуру, поведінку та зв'язки з іншими класами. Графічно цей тип представлений у вигляді прямокутника, який можна далі розділити на горизонтальні лінії. Ці частини можуть визначати назву класу, атрибуту та операції. Приклад реалізації даної діаграми наведено на рисунку 2.4.

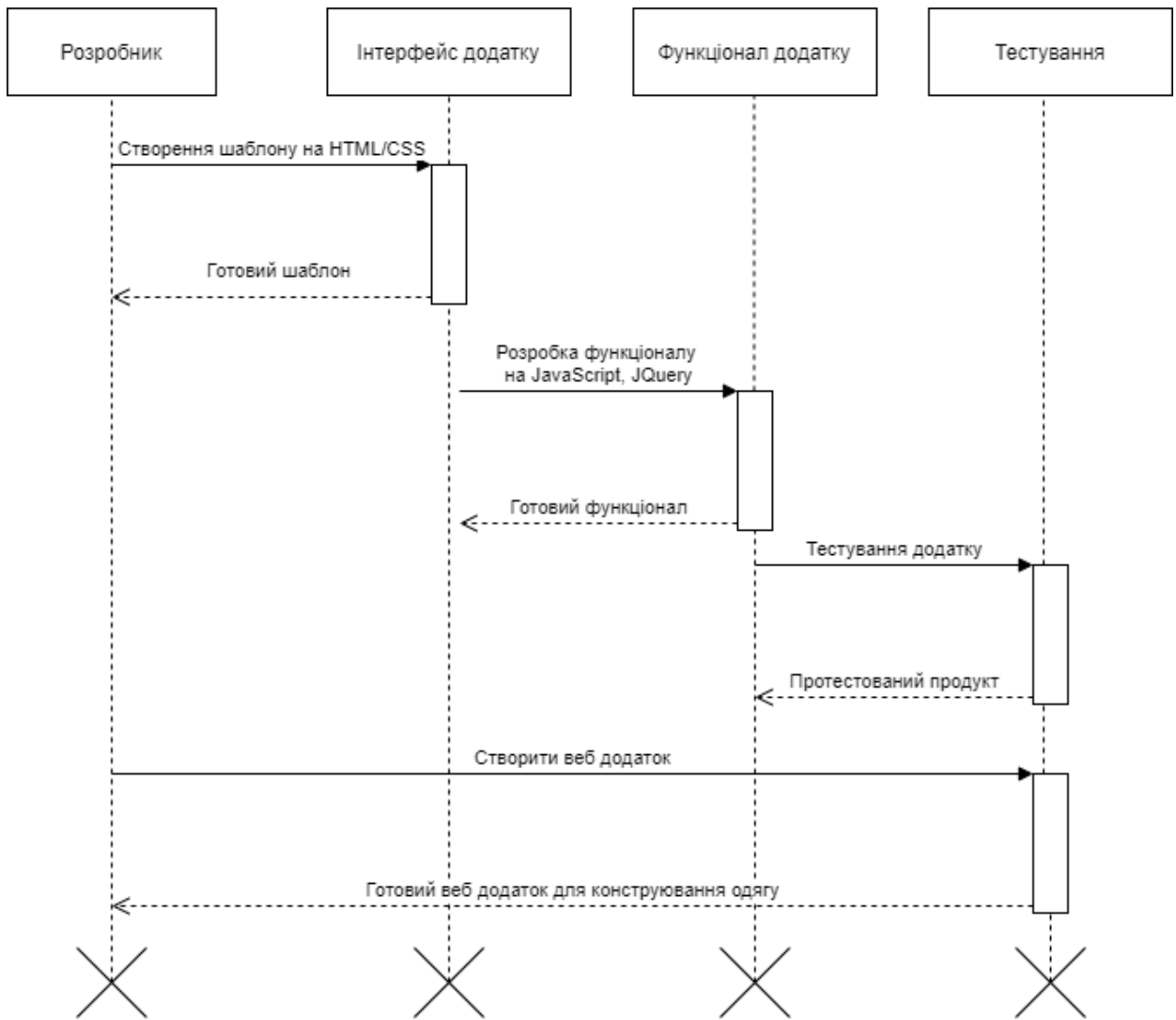


Рисунок 2.4 – Діаграма послідовності

Діаграма комунікацій – це спеціальна діаграма взаємодії, яка фокусується на обміні даними між різними учасниками взаємодії [16].

Діаграма комунікацій не потребує малювати кожного учасника, як діаграма послідовності та відображати послідовність повідомлень вертикально. Натомість учасників можна розміщувати за бажанням, дозволяючи комунікації показувати взаємозв'язок учасників та використовувати номери для представлення послідовностей повідомлень. Результат побудови діаграми показано на малюнку 2.5. Більш детально про етапи планування проекту наведено в додатку Б.

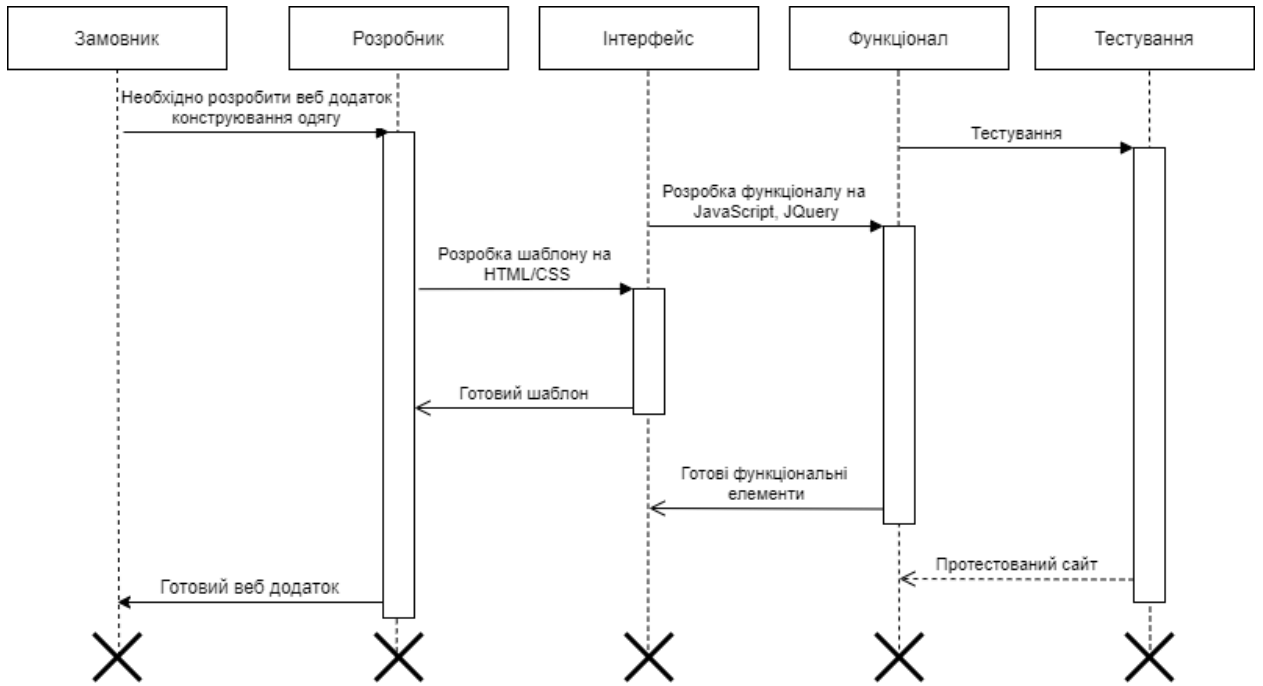


Рисунок 2.5 – Діаграма комунікації

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектура веб додатку

Веб додаток повинен складатися з наступних розділів:

- Головна сторінка містить розділи «Верхній одяг» та «Сукні», також кнопки для огляду спереду або ззаду.
- Розділ Пальто – містить підрозділи «Верх», «Низ», «Рукава», «Центр», тощо.
- Підрозділ «Верх», «Низ», «Рукава», «Центр», тощо – у кожному з них фотографії для конструювання з ціною.
- Розділ Сукні – містить підрозділи «Верх», «Низ», «Рукава».
- Підрозділ «Верх», «Низ», «Рукава» – у кожному з них фотографії для конструювання з ціною.

Форма Замовлення – містить форму для оформлення замовлення.

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і підрозділів. Наявність навігації на всіх сторінках сайту. Правильна структура інформації дозволяє користувачам без побоювання продовжувати дослідження сайту, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

3.2 Програмна реалізація

Для вирішення поставленої завдання було визначено ряд задач:

- Розробка прототипу сайту.
- Створення графічних елементів сайту.
- Створення шаблону інтерфейсу.

- Розробка функціональних елементів.
- Розробка адміністративної панелі.
- Тестування роботи додатку.

3.2.1 Розробка прототипу додатку

Створення веб-ресурсу – це комплекс заходів, який об'єднує в собі розробку дизайну, інформаційне наповнення, застосування веб-технологій, спрямованих на задоволення потреб відвідувачів і власників майбутнього сайту.

Після затвердження постановки завдання на розробку веб-сайту починається розробка дизайну. Орієнтуючись на технічне завдання був розроблений ескіз головної сторінки (рис. 3.1) та адміністративної панелі (рис. 3.2).

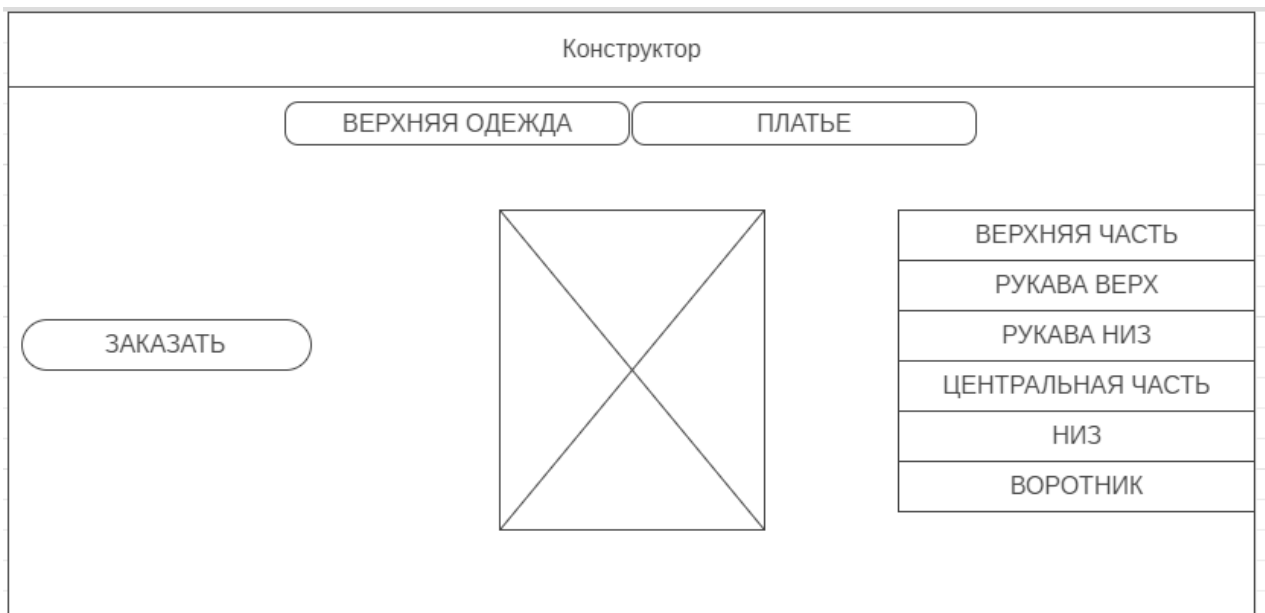


Рисунок 3.1 – Макет сайту

Рисунок 3.2 – Макет адміністративної панелі

Моделювання інтерфейсу додатку відбувалось за допомогою веб додатку wireframe. Wireframes – це додаток для організації повного функціоналу кінцевого продукту, у вигляді структури з поданням елементів інтерфейсу і навігації, їх взаємодії один з одним. Як правило, в wireframes не використовується типографіка, колір і будь-які інші графічні елементи оформлення, так як основна увага приділяється функціональності, поведінки і змісту. Wireframes фокусується на тому, як екрани взаємодіють між собою, а не на те, як вони виглядають [17].

В результаті виконання цього етапу роботи було отримано так звану "нарізку веб-сайту", тобто, готовий шаблон, який буде використаний як основа для розробки окремих сторінок.

3.2.2 Розробка додатку

Сама розробка буде відбуватись у безкоштовному текстовому редакторі VS Code. Visual Studio Code – це крос-платформенний редактор скриптів, створений корпорацією Майкрософт. Поряд з розширенням PowerShell він надає широкі інтерактивні можливості редагування скриптів, спрощуючи написання надійних скриптів PowerShell [18].

Також для збереження вихідного коду, а також відслідковування версій сайту було використано систему контролю версій Git. Системи контролю версій дозволяють розробникам зберігати всі зміни, внесені в код. Тому в разі помилки, вони можуть просто відкотити код до робочого стану замість того, щоб витратити години на пошуки маленької помилки або помилок, які ламають весь код [19].

Системи контролю версій також дають можливість декільком розробникам працювати над одним проектом і зберігати внесені зміни, щоб переконатися, що всі можуть стежити за тим, над чим вони працюють.

Написання сайту починається із створення «скелету» з використанням мови розмітки HTML. Після чого за допомогою мови каскадних стилів CSS формується зовнішній вигляд веб-сайту і надається візуальна форма елементів для зручного користування [20].

Інтерфейс сайту складається із таких елементів:

- Навігаційне меню для вибору частини одягу і ціни.
- Меню для вибору типу одягу.
- Загальна сума замовлення.
- Зображення створеного одягу.
- Форма замовлення.

В результаті, за зразком розробленого макету, було розроблено інтерфейс сайту та адміністративної панелі для конструювання одягу (рис. 3.3, 3.4). Приклад коду наведено в додатку В.

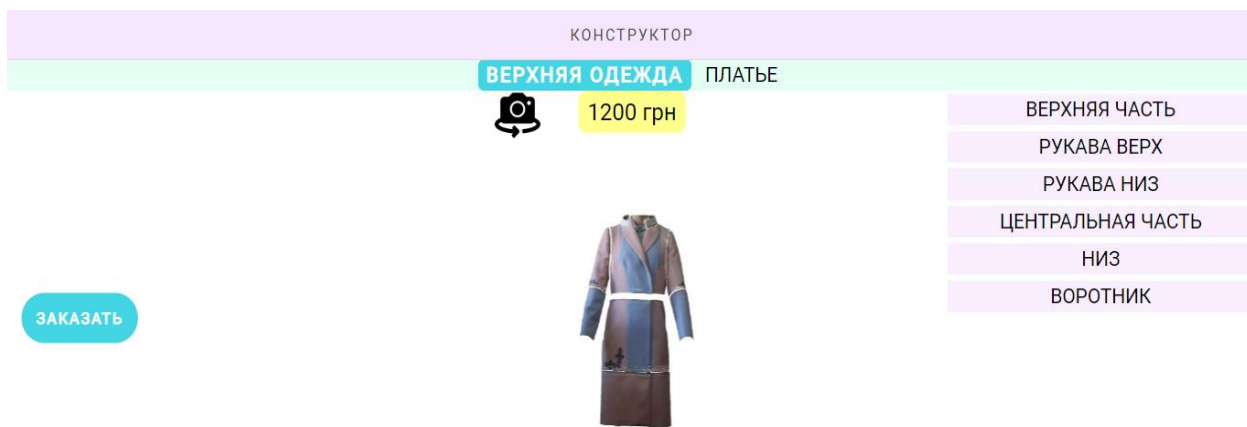


Рисунок 3.3 – Интерфейс додатку

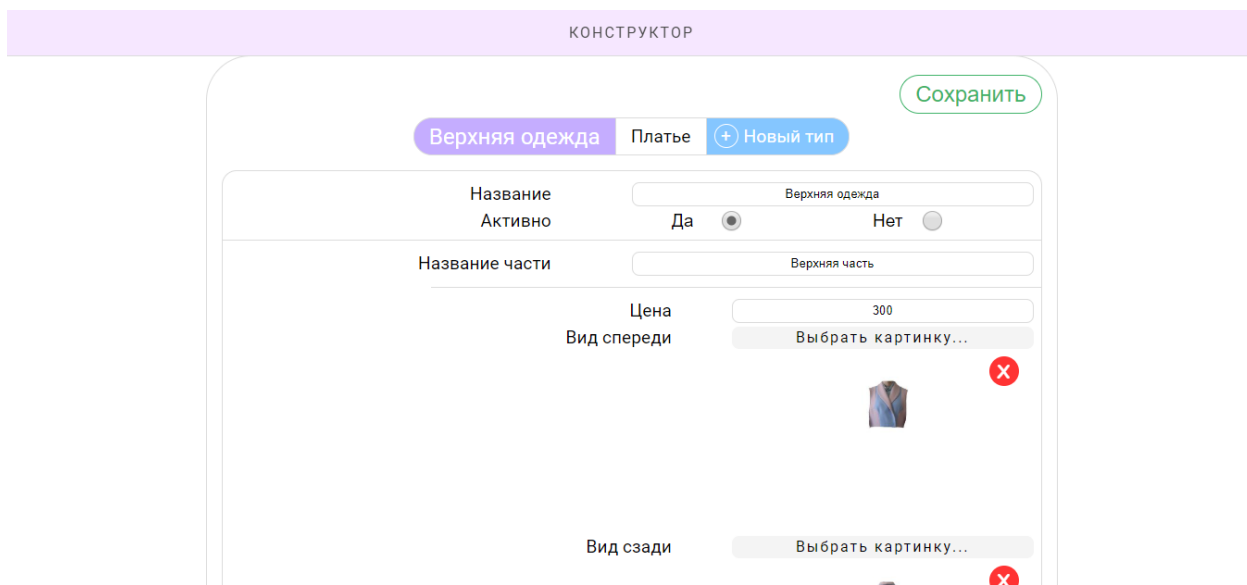


Рисунок 3.4 – Адміністративна панель додатку

Для надання сайту функціональності та інтерактивності було використано мови програмування PHP та JavaScript з використанням популярної бібліотеки jQuery. Приклад коду наведено в додатку В.

3.2.3 Тестування

Тестуванням – це процес виконання програми з метою перевірки правильності результатів її роботи. Основний метод для виявлення помилок в програмі – проведення випробувань. Випробуванням є процес виконання програми з метою перевірки правильності результатів її роботи і відповідності заданим характеристикам. Процес випробувань включає навмисне конструювання складних наборів вхідних даних, що створюють найбільші можливості для відмови програми.

Для тестування було обрано сучасні і найпопулярніші веб браузеры, а саме Google Chrome, Opera, Microsoft Edge, Internet Explorer 11. В результаті тестування кожен браузер відобразив сайт коректно з використанням усіх функцій сайту.

3.3 Використання програмного додатку

Даний дипломний проект, а точніше веб-сайт, що розроблений для конструювання одягу, складається з двох частин. А саме головна сторінка з можливістю конструювання одягу і замовлення та адміністративної панелі з можливістю додання і редагування продукції сайту.

3.3.1 Головна сторінка сайту

Головна сторінка веб-сайту складається з навігаційного меню де можна обирати частину одягу та ціну, меню для вибору типу одягу, зображення сконструйованого одягу і форму замовлення (рис. 3.5).

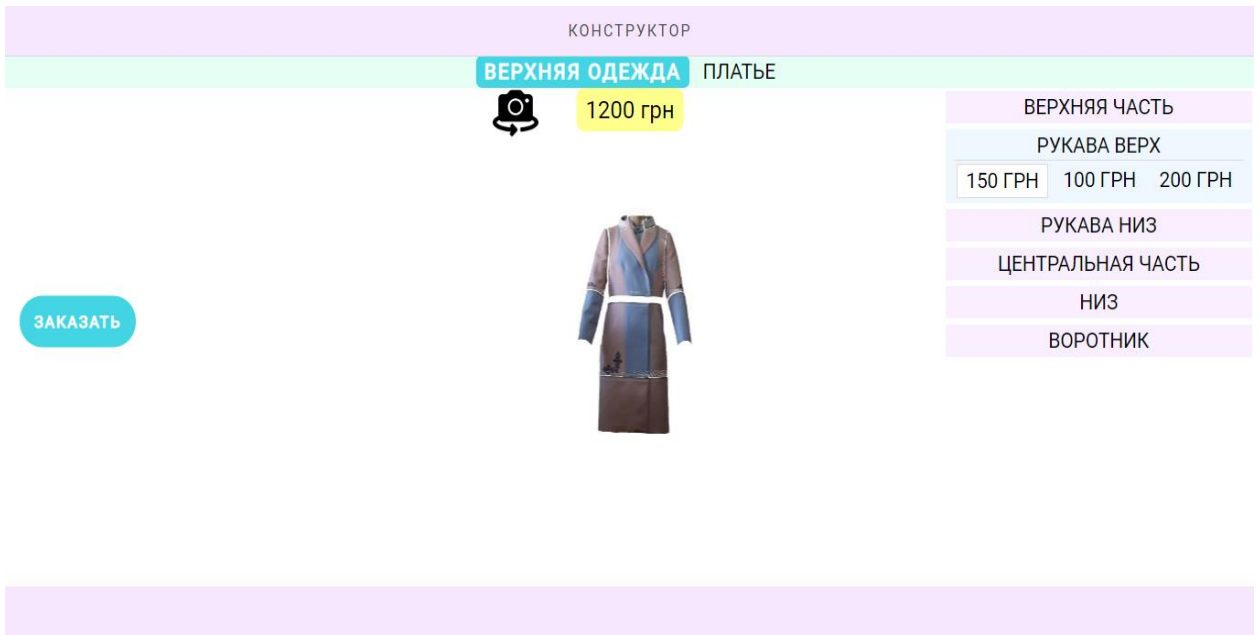


Рисунок 3.5 – Головна сторінка сайту

При натисканні на одну з частин одягу (в правій частині сайту) з'явиться список з цінами. Після вибору бажаної ціни на зображенні сконструйованого одягу зміниться та частина яку було попередньо обрано. Також зміниться загальна сума над зображенням одягу (рис. 3.6).



Рисунок 3.6 – Приклад конструювання одягу

Для перегляду задньої частини одягу потрібно натиснути кнопку у вигляді фотокамери біля загальної ціни товару (рис. 3.7).



Рисунок 3.7 – Вигляд одягу ззаду

Для замовлення сконструйованого одягу було додано кнопку «Заказать» при натисканні на яку відкриється модальне вікно з формою для ведення даних користувача (рис. 3.8).

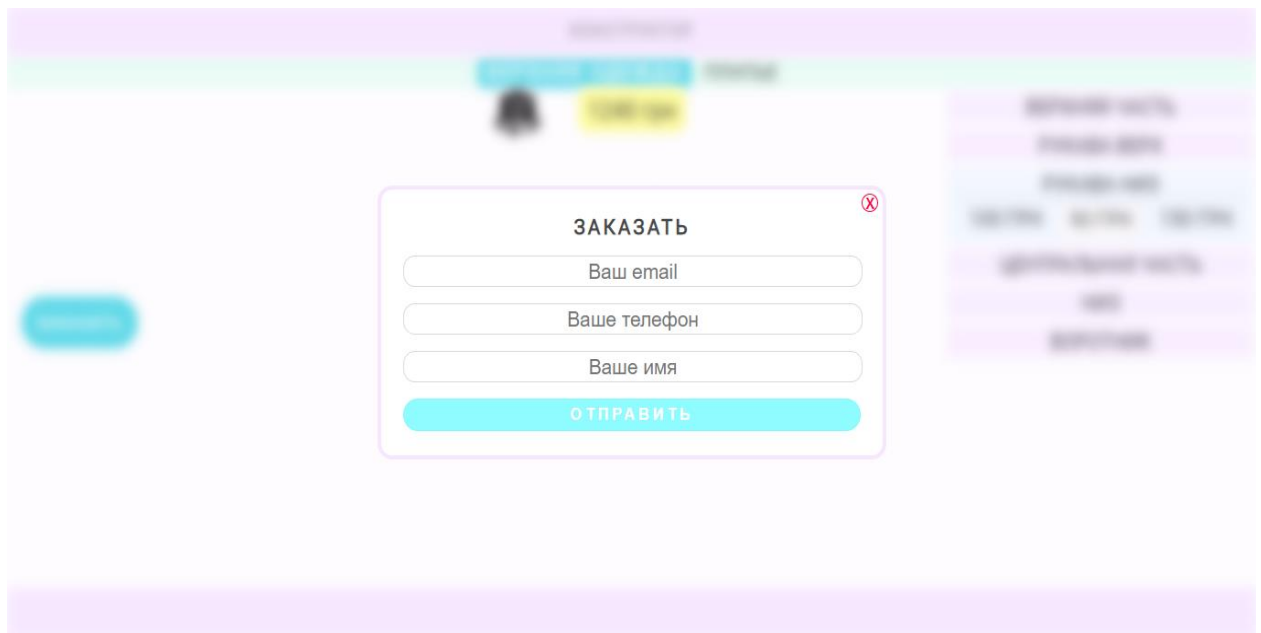


Рисунок 3.8 – Форма замовлення

Після введення коректних даних від користувача потрібно натиснути на кнопку «Отправить» після чого з'явиться повідомлення про успішно оформлене замовлення і посиланням на головну сторінку (рис. 3.9).

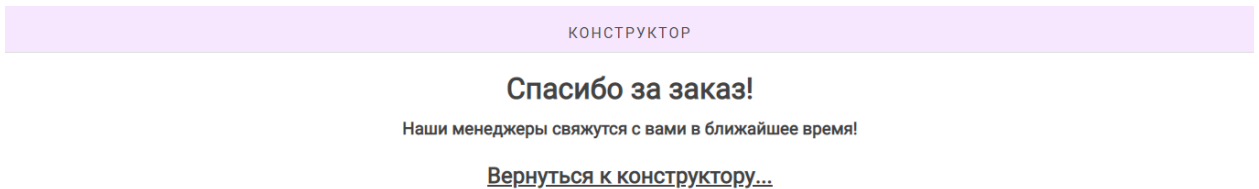


Рисунок 3.9 – Повідомлення про успішне замовлення

3.3.2 Адміністративна панель

Адміністративна панель являється закритою частиною сайту доступ до якої мають лише адміністратори перейшовши за спеціальним посиланням. Адміністративна панель складається з головного меню, функціональних кнопок для додання, збереження та видалення продукції сайту (рис. 3.10).

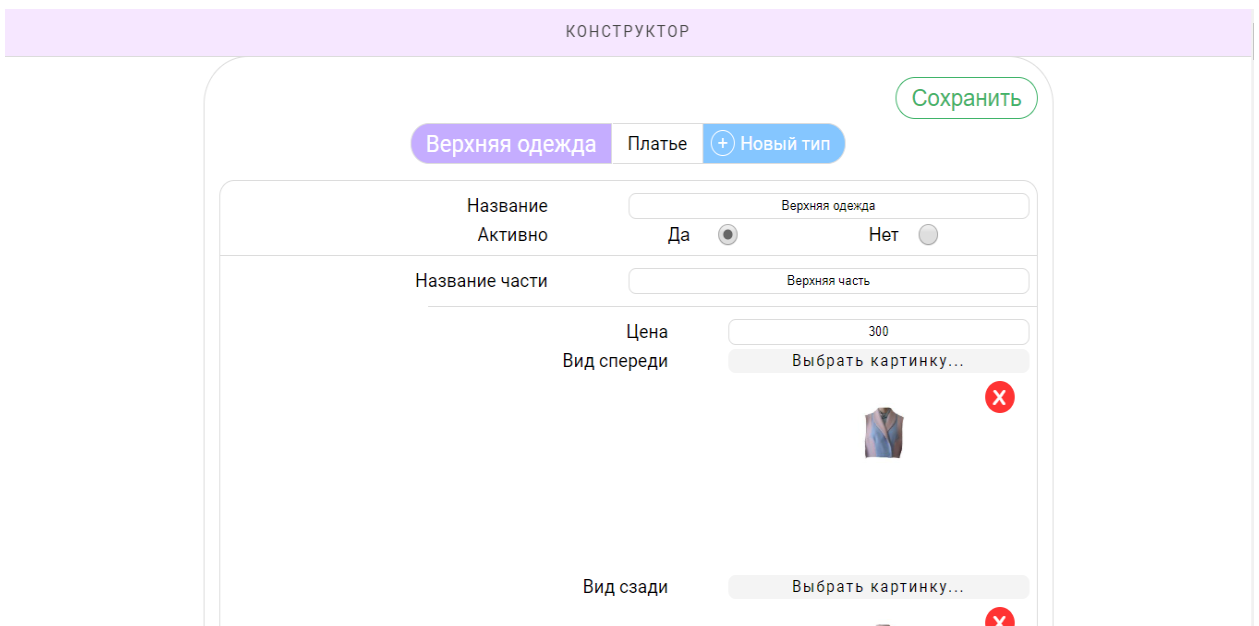


Рисунок 3.10 – Адміністративна панель

Меню складається з кнопок для перемикання між типами одягу і кнопки для створення нового типу. При натисканні на кнопку «Новий тип» з’явиться форма в яку необхідно ввести назву типу одягу та визначити чи буде даний тип доступний для користувачів сайту. За необхідністю можна видалити даний тип натиснувши відповідну кнопку.

Рисунок 3.11 – Додання нового типу одягу

Щоб даний тип одягу розбити на частини які будуть відрізнятись зовнішнім виглядом та ціною потрібно натиснути кнопку «Добавить часть» після чого з'явиться форма в яку потрібно вказати назву частини одягу. Потім натиснути кнопку «Добавить предмет», що дасть можливість додавати частини одягу і вказувати ціну на них (рис. 3.12). Також є можливість видалення зображень і предметів.

Рисунок 2.12 – Додання предмету одягу

Після завершення додання інформації про тип одягу необхідно натиснути на кнопку «Сохранить» після чого всі данні збержуться і з'являться на сайті.

Матеріали роботи були представлені на конференції ІМА-2020 [21].

ВИСНОВОК

Головним завданням проекту є досягнення всіх цілей та виконання завдань проекту, а саме створення веб-додатку для конструювання одягу. За допомогою проекту можна створити власний унікальний дизайн одягу і замовити доставку не виходячи з дому.

При виконанні даної роботи було проведено аналіз предметної області на основі літературних джерел, визначено актуальність роботи. Проведено аналіз аналогів для визначення недоліків і переваг конкурентів.

В процесі роботи досліджена предметна область. Оптимізовано web-інтерфейс і навігація сайту, для того щоб користувачеві було зручніше орієнтуватися в віртуальному просторі.

Було сформульовано технічне завдання до дипломної роботи, в якому описується призначення сайту, мета створення, цільова аудиторія використання. Підготовлена документація по планування робіт.

Проект реалізований з використанням бібліотеки jQuery, основою якого являється мова програмування JavaScript.

Дана робота допоможе всім бажаючим створити власний дизайнерський одяг. Також дана розробка буде корисною як початківцям так і кваліфікованим дизайнерам.

Отже, проект буде надавати повний функціонал для конструювання одягу. Опираючись на всю інформацію та знання був розроблений веб-додаток для конструювання одягу. Всі цілі і завдання були досягнуті і виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке PHP? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/intro-what-is.php>. (дата звернення: 22.04.2020).
2. Девід Карр. The PHP Workshop: A New, Interactive Approach to Learning PHP, 2019. – 360 с.
3. Крокфорд Дуглас. JavaScript. Сильні сторони, 2016. – 176 с.
4. Хавербеке Марейн. Виразний JavaScript. 2 видання, 2015. – 745 с.
5. Ерл Каслдайн, Крейг Шарки Вивчаємо JQUERY. Новий стиль програмування на JavaScript Санкт-Петербург: Питер, 2015. – 366 с.
6. Що таке JSON [Електронний ресурс] – Режим доступу до ресурсу: <https://losst.ru/chto-takoe-json>. (дата звернення: 24.04.2020).
7. Джон Дукетт. HTML and CSS design and build websites, 2014. – 420 с.
8. Що таке CSS сайтів [Електронний ресурс] – Режим доступу до ресурсу: <http://phpist.com.ua/css/5-whatcss/>. (дата звернення: 22.04.2020).
9. Ерік А. Мейер CSS-каскадні таблиці стилів. Детальний керівництво Москва: Символ, 2016. – 572 с.
10. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5, 2016. – 510 с.
11. А. Кириченко, А. Хрусталева. HTML5 + CSS3. Основи сучасного WEB-дизайну 2018. – 346с.
12. Крістіан Даруй, Богдан Брінзаре, Філіп Черчез-Тоза, Міхай Бусика Ajax і PHP. Розробка динамічних веб-додатків. Москва: Символ, 2017. – 332 с.
13. Люк Веллінг, Лора Томсон Розробка Web-додатків за допомогою PHP і MySQL Москва: Вільямс, 2015. – 876 с.
14. Діаграми UML [Електронний ресурс] – Режим доступу до ресурсу: <https://planerka.info/item/diagrammy-kommunikacij-uml/>. (дата звернення: 23.04.2020).

15. Теорія та практика UML [Електронний ресурс] – Режим доступу до ресурсу: http://www.it-gost.ru/articles/view_articles/94. (дата звернення: 23.04.2020).
16. The 2020 Roadmap To Fullstack Web Development [Електронний ресурс] – Режим доступу до ресурсу: <https://codingthesmartway.com/the-2020-roadmap-to-fullstack-web-development/>. (дата звернення: 22.04.2020).
17. Best Personal and Niche Blogs (30+ Real Examples) [Електронний ресурс] – Режим доступу до ресурсу: <https://firstsiteguide.com/examples-of-blogs/>. (дата звернення: 22.04.2020).
18. Безкоштовні редактори коду для програмістів [Електронний ресурс] – Режим доступу до ресурсу: <https://techrocks.ru/2018/09/10/top-6-popular-free-source-code-editors/>. (дата звернення: 24.04.2020).
19. Документація Git [Електронний ресурс] – Режим доступу до ресурсу: <https://git-scm.com/>. (дата звернення: 24.04.2020).
20. Хеслоп П. HTML самого початку. С.-Пб: Санкт-Петербург, 2014. – 450с.
21. Ковенко. А. С. Веб-додаток для конструювання одягу / Ковенко Анна Станіславівна // ІМА-2020 / Ковенко Анна Станіславівна. – Суми, 2020. – (Сумський державний університет).

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

ТЕХНІЧНЕ ЗАВДАННЯ на розробку Web-сервісу конструювання одягу

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ВЕБ-ДОДАТКУ

1.1 Призначення веб-додатку

Веб-додаток призначений для того, щоб кожен бажаючий мав можливість розробити власний дизайн одягу і замовити не виходячи з дому.

1.2 Мета створення веб-додатку

Метою дипломного проекту є створення веб сервісу конструювання одягу для пошивку за власним зразком.

1.3 Цільова аудиторія

Цільова аудиторія – це група людей, якій потенційно потрібна послуга або товар. У всіх членів цієї групи є спільні ознаки.

Цільова аудиторія даного веб-додатку:

- Дизайнери початківці.
- Професійні дизайнери.
- Кожен бажаючий створити власний дизайн одягу.

2 ВИМОГИ ДО ВЕБ-ДОДАТКУ

2.1 вимоги до структури й функціонування веб-додатку

Веб-додаток повинен бути реалізований у вигляді сайту, доступного в мережі Інтернет. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.2 вимоги до персоналу

Для підтримки сайту і експлуатації веб-інтерфейсу від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером.

2.3 вимоги до стилістичного оформлення сайту

Інтерфейс сайту повинен забезпечувати інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, логічний перехід до розділів і сторінок. Посилання на сторінки повинні бути забезпечені заголовками. При виборі будь-якого з пунктів меню користувачем повинна завантажуватися відповідна йому інформаційна сторінка, а в блоці меню відкриватися список підрозділів.

3 ОСНОВНІ ВИМОГИ

3.1 структура веб-додатку

Веб-додаток повинен складатися з наступних розділів:

- Головна сторінка містить розділи «Верхній одяг» та «Сукні», також кнопки для огляду спереду або ззаду.
- Розділ Пальто – містить підрозділи «Верх», «Низ», «Рукава», «Центр», тощо.
- Підрозділ «Верх», «Низ», «Рукава», «Центр», тощо – у кожному з них фотографії для конструювання з ціною.
- Розділ Сукні – містить підрозділи «Верх», «Низ», «Рукава».
- Підрозділ «Верх», «Низ», «Рукава» – у кожному з них фотографії для конструювання з ціною.
- Форма Замовлення – містить форму для оформлення замовлення.

3.2 навігація

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і підрозділів. Наявність навігації на всіх сторінках сайту. Правильна структура інформації дозволяє користувачам без побоювання продовжувати дослідження сайту, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

3.3 вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 7.5 і вище, або Opera 9.5 і вище, або Safari 6.1 і вище, або Chrome 7 і вище.
- Включена підтримка javascript і Flash.

3.4 функціональні вимоги

- Перегляд доступних фотографій.
- Конструювання одягу.
- Перегляд цінової політики.
- Зворотній зв'язок.
- Замовлення онлайн.

3.5 наповнення сайту (контент)

Первинна розробка та верстка контенту (інформаційного вмісту) сайту повинна проводитися силами Виконавця за безпосередньої участі Замовника. Замовник надає всі необхідні Виконавцю текстові та графічні матеріали, а також коментарі щодо їх змісту, обсягу, оформлення і розміщення.

ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

1 ІДЕНТИФІКАЦІЯ ІДЕЇ ПРОЕКТУ

Мета даного сайту в наданні потенційним клієнтам можливості самостійно створити і змодельовати власний одяг використовуючи зручний інтерфейс, а також замовити змодельований одяг.

Дипломний проект призначений для того, щоб користувач міг з легкістю змодельовати власну модель одягу і замовити не виходячи з дому.

Веб додаток повинен бути реалізований у вигляді сайту, доступного в мережі Інтернет. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2 ДЕТАЛІЗАЦІЯ МЕТИ МЕТОДОМ SMART

Технологія SMART – це сучасний метод встановлення робочих цілей. Інтелектуальна система встановлення цілей дозволяє нам узагальнити всю наявну інформацію, встановити прийнятний робочий час, визначити достатність ресурсів та надати чіткі, точні та конкретні завдання для всіх учасників етапу встановлення цілей.

SMART є аббревіатурою, значення якої: Specific, Measurable, Achievable, Relevant, Time bound. Кожна буква аббревіатури SMART означає критерій ефективності поставлених цілей. Розглянемо кожен критерій smart мети більш докладно.

Результати деталізації методом SMART розміщені у табл. А.2.1.

Таблиця А.2.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити сайт для конструювання одягу з урахуванням поставлених цілей і вимог замовника.
Measurable (вимірювана)	Використовуючи мінімум ресурсів розробити якісний програмний продукт.
Achievable (досяжна)	Реалізація сайту здійснюється за допомогою мов програмування PHP і JavaScript.
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з календарним планом.

3 ОПИСАННЯ ФАЗИ РОЗРОБКИ ІТ—ПРОЕКТУ

3.1 Планування змісту структури робіт ІТ—проекту (WBS)

Структура розбивки на роботу – це інструмент, який можна використовувати для розділення проекту на кілька частин. Він встановлює відділ ієрархічної структури реалізації проекту для всіх відповідних працівників.

Побудови WBS проекту завжди розбивається на підпроекти, різні рівні роботи та детальні робочі пакети. Розподіл – це розподіл результатів проекту на менші, більш керовані компоненти пакету. Робочі пакети зазвичай відповідають найнижчому рівню деталізації і складаються з однієї роботи. Розподіл має бути правильним, тобто будь-який елемент на рівні WBS повинен бути достатнім для створення відповідного вищого елемента.

Ієрархічна структура роботи – це в основному перелік завдань проекту. Він може відображатися графічно або у формі описів, що містяться у відображеному результаті. Ієрархічна структура роботи організовує та визначає весь зміст проекту. Роботи, не включені до WBS, не є роботами проекту.

Буде побудовано структуру WBS, в якій перераховані всі виконані роботи відповідно до основного етапу. Діаграма WBS показана на малюнку Б.3.1:

1. Формування технічного завдання — розробка технічного завдання, що встановлює основне призначення, показники якості, техніко-економічні та спеціальні вимоги до розроблюваного інструментального засобу. Формування технічного завдання включає в себе підпункти (визначення предметної області, призначення програмного продукту, визначення мови написання, визначення цільової аудиторії, визначення вимог до програмного продукту).

2. Розробка програмного продукту – написання відповідних модулів, що забезпечують функціонування програмного продукту.

2.1 Визначення проблеми.

2.2. Розробка інтерфейсу програмного продукту.

2.3. Розробка стилю засобами CSS.

2.4. Наповнення контентом.

2.5. Додання інтерактивних елементів.

3. Тестування – перевірка роботи програмного продукту, виявлення

ПОМИЛОК.

WBS-структура для даного проекту представлена на рисунку Б.3.1.



Рисунок Б.3.1 – WBS-структура Веб додатку для замовлення одягу

3.2 Планування структури організації готового проекту

Структура проекту OBS – організаційна структура виконавця проекту (організації). Вона визначається переліком робочих пакетів нижчого рівня для кожної гілки структури WBS. Представляти особу, відповідальну за виконання робочого пакету (відповідальна особа – це не обов'язково особа, яка відповідає за організацію (відділ), а особа, яка безпосередньо організовує виконання робіт)

Організаційна структура – це графічне зображення учасників проекту та осіб, відповідальних за реалізацію проекту. Найвищий рівень OBS – команда проекту.

На наступному рівні фіксуються виконавці: організації, відділи тощо. Потім, рівнем нижче, для кожного виконавця вказують прізвища конкретних осіб, які будуть відповідати за виконання елементарних робіт WBS. Потрібно пам'ятати, що відповідальні – це не обов'язково керівники, а ті співробітники, які безпосередньо організують і відповідають у виконавця за виконання елементарної роботи, зазначеної у WBS. Для них ця елементарна робота також є проектом (у порівнянні з загальним проектом). Для себе вони також можуть побудувати WBS — структуру й застосовувати інші інструменти планування.

Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.3.2.



Рисунок Б.3.2 – Організаційна структура виконавців

3.3 Побудова матриці виконавців пакетів робіт

На підставі OBS та WBS структур було створено список виконавців, що функціонують в проекті.

Таблиця Б.3.1 – Виконавці проекту

Фази	Виконавці	
	Ковенко А.	Кузнєцов Е.Г.
Аналіз предметної області		+
Формування ТЗ	+	
Мета проекту		+
WBS-структура	+	
Календарний план проекту	+	
Управління ресурсами	+	
Управління ризиками	+	
Управління якістю	+	
Розробка сайту	+	
Тестування	+	
Інструкція користувача	+	
Здача в експлуатацію		+

4 ПОБУДОВА КАЛЕНДАРНОГО ГРАФІКУ ВИКОНАННЯ ПРОЕКТУ

Діаграма Ганта – це горизонтальна лінійна діаграма, наведені вище завдання проекту представлені довгими сегментами і характеризуються датами початку та закінчення, затримками та, можливо, іншими параметрами часу. Щоб по-справжньому зрозуміти тривалість роботи, виходячи з обмеженого використання ресурсів, виходячи з частини мережевої моделі та всього проекту (включаючи вихідні та святкові дні), був розроблений календарний графік роботи.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рисунку Б.4.1, Б.4.2.

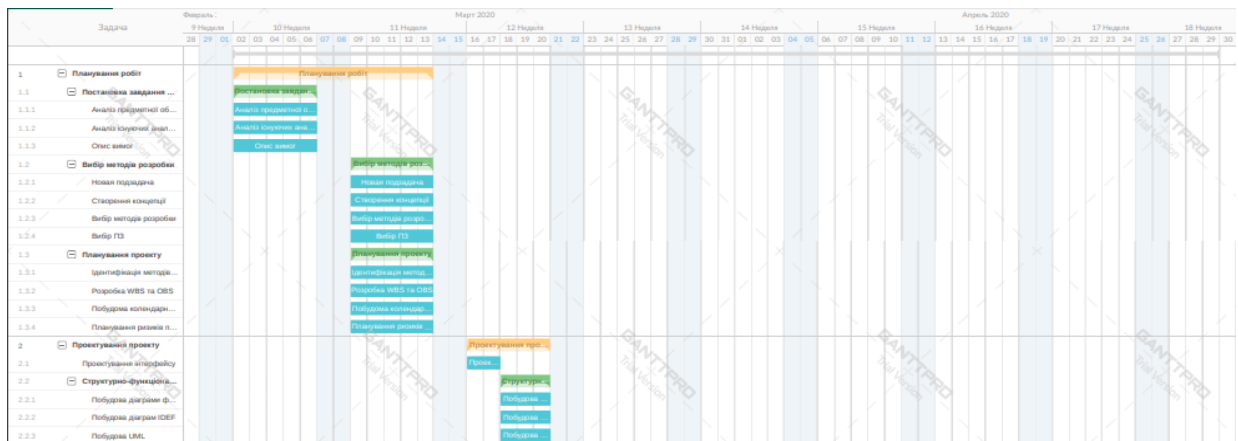


Рисунок Б.4.1 – Діаграма Ганта

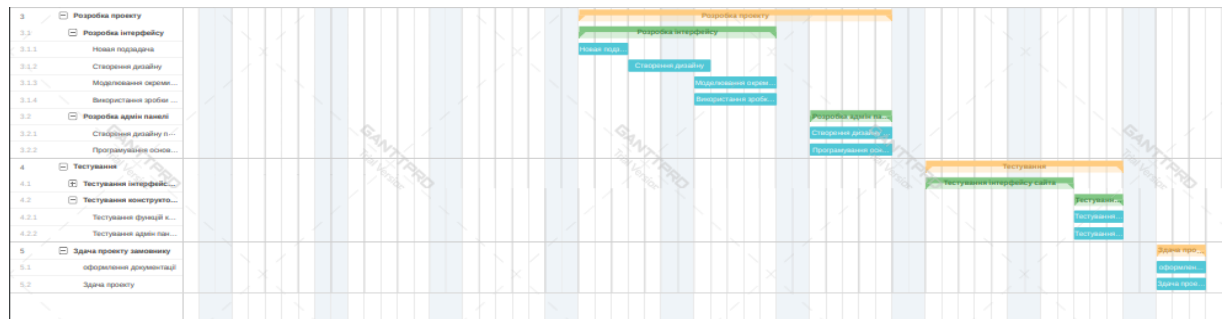


Рисунок Б.4.2 – продовження діаграми Ганта

5 ІДЕНТИФІКАЦІЯ РИЗИКІВ

Якісна та кількісна оцінка трудових ризиків. У якісній оцінці ми визначаємо ризики, які потребують швидкого реагування. Оцінка визначить ступінь важливості ризику та дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде проведена для кращого визначення ризиків та їх впливу на реалізацію проекту. Кількісну та якісну оцінку ризику можна використовувати окремо або разом, залежно від наявного часу та бюджету та необхідності кількісної чи якісної оцінки ризику.

Ризик – це ймовірнісна подія, яка у випадку своєї появи негативно або позитивно впливає на проект.

Управління ризиками – це процес реагування на події та зміни ризиків під час реалізації проекту. Важливий моніторинг ризиків.

Моніторинг ризиків включає контроль ризику протягом життєвого циклу проекту. Моніторинг ризику якості забезпечує управління інформацією, щоб допомогти вам прийняти ефективні рішення до того, як виникнуть ризикові події.

Найпоширенішими характеристиками ризику є загроза або небезпека відмови певної діяльності, ризик несприятливих наслідків, зміни навколишнього середовища, які можуть спричинити втрату ресурсів, втрати та ризики, які слід покрити.

Процес управління ризиками включає в себе такі пункти:

- 1) Ідентифікація ризиків (виявлення ризиків).
- 2) Оцінювання ризиків (оцінка ймовірності та впливу).
- 3) Заходи реагування на ризики.
- 4) Моніторинг ризиків.

Ідентифікація ризиків – виявлення ризиків здатних вплинути на проект, документальне оформлення їх характеристик.

В даному випадку на першому етапі, в процесі виявлення ризиків можна виділити ряд ризиків.

Зовнішні непередбачувані ризики:

- Природні катастрофи: повені; землетруси; шторми; кліматичні катаклізми тощо.
- Злочини: вандалізм; саботаж; тероризм.
- Неочікувані зовнішні ефекти: екологічні; соціальні.
- Зриви: у створенні необхідної інфраструктури; через банкрутство підрядників; у фінансуванні: через помилки у визначенні цілей проекту; через неочікувані політичні зміни.

Зовнішні передбачувані (проте не визначені) ризики:

- Ринковий ризик у зв'язку із: зміною вимог споживачів; економічними змінами; посиленням конкуренції; втратою позицій на ринку.
- Операційні ризики: порушення безпеки; відхилення від цілей проекту; неможливість підтримання умов роботи елементів проекту. неприйнятний вплив на навколишнє середовище; негативні соціальні наслідки; зміни курсу валюти, незрозуміла інфляція.
- Внутрішні ризики. Внутрішні організаційні ризики.
- Зриви планів робіт через: недостачу робочої сили; нестачу часу. помилки проектування; помилки планування; недоліки координації робіт; зміни керівництва; конфлікти та саботаж; зміну можливостей замовника проекту. недостатнє управління.
- Перевитрати коштів через: зриви планів робіт; невірну стратегію. некваліфікований персонал; переплати за роботу/матеріали; неузгодження частин проекту; невірний кошторис.
- Внутрішні технічні ризики.
- Зміни технології.
- Специфічні ризики технології, що закладаються до проекту.
- Помилки в проектно-кошторисній документації.

Інші ризики:

- Прямі втрати майна: по транспортних спорах; обладнання і т.д.

- Непрямі збитки: перестановка обладнання; неотримання орендного прибутку.
- Порушення запланованого ритму діяльності; збільшення необхідного фінансування.
- Ризики, що підлягають обов'язковому страхуванню: нещасні випадки на виробництві; пошкодження майна; викрадення транспортних засобів.

За джерелами виникнення:

- Неправильна оцінка ринкової ситуації (ємність ринку, рівень конкуренції).
- Невизначеність мети, інтересів і поведінки учасників проекту, проблеми управління командою.
- Виробничо-технологічне устаткування, виробничі ризики (аварії, брак).
- Неточність проектної документації (витрати, строки реалізації проекту, технічні і технологічні параметри).
- Ризик зміни пріоритетів розвитку підприємства і підтримки з боку керівництва.
- Неповнота або неточність інформації про фінансовий стан та ділову репутацію учасників.

Систематичний ризик – визначається зовнішніми обставинами, не залежить від суб'єкта та не регулюється ним:

- Ризики, пов'язані з нестабільністю нормативного поля режиму інвестування.
- Можливість змін природно-кліматичних умов, стихійного лиха.
- Можливість погіршення політичної ситуації, несприятливі соціально-політичні зміни.
- Коливання ринкової кон'юнктури, валютного курсу.

За сферами прояву:

- Економічні, пов'язані із зміною економічних факторів.
- Політичні, пов'язані із зміною політичного курсу країни.
- Соціальні, пов'язані з соціальними складностями.
- Екологічні, пов'язані з екологічними катастрофами.

За видами втрат (наслідками):

- Трудові втрати – втрати фондів робочого часу.
- Перевитрати часу – уповільнення процесу реалізації проекту у порівнянні з планом.
 - Фінансові втрати – пов'язані з непередбаченими виплатами (штрафи, уплата додаткових податків), неотриманням коштів з передбачених джерел.
 - Перевищення витрат – виникає внаслідок зміни початкового плану реалізації проекту або зниження розрахункових витрат на будівництво, затримання строків будівництва.
 - Соціальні втрати – шкода здоров'ю та життю людей, довкіллю, престижу організації, іміджу учасників.
 - Ризик нежиттєздатності проекту – передбачені доходи від проекту можуть бути недостатніми для покриття усіх видів витрат.
 - Ризик недоплати заборгованостей – тимчасове зниження доходу через короткострокове падіння попиту на продукт, або зниження ціни на нього.
 - Податковий ризик – неможливість використання податкових пільг, встановлених законодавством; зміна податкового законодавства.
 - Виробничі ризики – виникають після завершення впровадження проекту; пов'язані з проблемами технічної підготовки і розробки проекту, поганого інжинірингу, незадовільного навчання персоналу, нестачі сировини і зростання витрат виробництва.

В процесі аналізу для визначення числових значень ймовірності виникнення ступеня впливу, зазвичай застосовується метод експертних

оцінок. На їх основі визначається ранг ризику, як потенційний вплив ризику на проект, який оцінюється як добуток ймовірності виникнення та ступеню впливу.

Шкала оцінки ризику може відповідати емпіричній шкалі оцінки ризику:

5 балів – критичний ризик;

4 бали – максимальний ризик;

3 бали – високий ризик;

2 бали – нормальний ризик;

1 бал – малий ризик.

$RV = P * I$, де RV – ранг ризику; P – ймовірність виникнення; I – ступінь впливу.

Таблиця Б.5.1 – Ризики проекту

№	Risk description	Impact	probability	RV	Mitigation
1	Нестабільність Програмного забезпечення	5	3	С	Використовувати перевірене та якісне Програмне забезпечення
2	Некоректний функціонал веб-додатку	4	2	М	Тестування веб-додатку
3	Неактуальність веб-додатку	3	2	М	Використовувати найновіші тренди та сучасний підхід

Продовження таблиці Б.5.1

№	Risk description	Impact	probability	RV	Mitigation
4	Додаткові вимоги	4	3	M	Обговорити всі вимоги з замовником перед початком проекту
5	Затримка фінансування	3	1	L	Обговорити план фінансування проекту

- Зелений колір – прийнятні ризики.
- Жовтий колір – виправданні ризики.
- Червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.5.2.

Таблиця Б.5.2 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	1,8,11,12,13
2	Виправдані	$3 \leq R \leq 4$	2,4,6,10,15
3	Недопустимі	$6 \leq R \leq 9$	3,5,7,9,14

ДОДАТОК В. ПРОГРАМНА РЕАЛІЗАЦІЯ

1. Фрагмент коду з файлу `index.php`: відповідає за підключення всіх необхідних компонентів.

```
<?php
require_once "functions.php";
$config = loadConfig();
$title = $config['sitename'];
include "header.php";
echo renderConstructor();
include "footer.php";
```

2. Фрагмент коду з файлу `functions.php`: відповідає за всіх функцій роботи сайту, а саме обробка замовлень, створення нових моделей одягу.

```
<?php
function controlBlock($blocks) {
    $render = "";
    $render .= '<div class="controls">';
    foreach ($blocks as $block) {
        $render .= "<div class='control control-item' id='change_{".$block['id']."' data-
target='{.$block['id']}'><span>{".$block['name']}</span></div>";
    }
    $render .= '</div>';

    return $render;
}

function variantBlock($blocks)
{
    $render = "";
```

```

foreach ($blocks as $block) {
    $render .= "<div class='controls-select' data-select-id='{ $block['id'] }'><span class='select-
title'>{ $block['name'] }</span>";
    $controls = "";
    foreach ($block['variants'] as $key => $item) {
        $controls .= "<span data-target='$key'>$item[name]</span>";
        $render .= "<div class='select-item ' . ($key == 0 ? 'open' : ") . "' data-self='$key'><img
class='display-img'          src='catalog/{ $block['id'] }/{ $item['img'] }'          front-
src='catalog/{ $block['id'] }/{ $item['img'] }'          alt=''><span
class='price'>{ $item['price'] }</span></div>";
    }
    $render .= "<div class='sub-controls'>$controls</div>";
    $render .= '</div>';
}

return $render;
}

function backButton()
{
    $render = "";

    $render .= "<div id='button-rotate'><span class='open' data-state='front'>ЛИЦО</span><span
data-state='back'>СПИНА</span></div>";

    return $render;
}

function getAdmin()
{
    // todo^ priority of each part
    $render = "";
    $data = loadData();
    $render .=

```

```

"<div class='admin-content'>" .
"<div class='tmp-hidden'>" .

"<div class='js-item-template group in-item'>" .
"<label for='theIdReplace[part][theCurrent][items][theItemId][price]'>Цена</label>" .
"<input type='text' name='theIdReplace[part][theCurrent][items][theItemId][price]'>" .
"<label class='image-label' >" .
"<span class='the-label'>Вид спереди</span>" .
"<span class='the-action active'>Выбрать картинку...</span>" .
"<img src='//via.placeholder.com/200x200/' class='the-image'>" .
"<div class='remove-img'>X</div>" .
"<input type='file' name='theIdReplace[part][theCurrent][items][theItemId][image]'>" .
"<input
                                                    type='hidden'
name='theIdReplace[part][theCurrent][items][theItemId][image_is_set]' value='>' .
"</label>" .
"<label class='image-label' >" .
"<span class='the-label'>Вид сзади</span>" .
"<span class='the-action active'>Выбрать картинку...</span>" .
"<img src='//via.placeholder.com/200x200/' class='the-image'>" .
"<div class='remove-img'>X</div>" .
"<input type='file' name='theIdReplace[part][theCurrent][items][theItemId][image_b]'>" .
"<input
                                                    type='hidden'
name='theIdReplace[part][theCurrent][items][theItemId][image_b_is_set]' value='>' .
"</label>" .
"<span class='clickable remove'>Удалить предмет</span>" .
"</div>" .

"<div class='js-part-template group in-part'>" .
"<label for='theIdReplace[part][theCurrent][name]'>Название части</label>" .
"<input type='text' name='theIdReplace[part][theCurrent][name]'>" .
"<span class='clickable add' data-target='item' data-parent='theIdReplace' data-
current='theCurrent'>Добавить предмет</span>" .
"<span class='clickable remove'>Удалить часть</span>" .
"</div>" .

```

```

"<div class='js-type-template group active in-tab' id='theIdReplace'" .
"<label for='theIdReplace[name]'">Название</label>" .
"<input      class='action-title'      data-target='theIdReplace'      type='text'
name='theIdReplace[name]'">" .
"<label      for='theIdReplace[active]'">Активно</label><input      type='radio'
name='theIdReplace[active]' value='Да'">" .
"<input type='radio' name='theIdReplace[active]' value='Нет' checked>" .
"<span class='clickable add' data-target='part' data-parent='theIdReplace'">Добавить
часть</span>" .
"<span class='clickable remove'">Удалить тип</span>" .
"</div>" .

"</div>" .

"<form action='admin.php' method='post' enctype='multipart/form-data'"><div class='global-
controls'">
  <input class='clickable' type='submit' name='nonce' value='Сохранить'">
</div>" .
  renderTabs($data) .
  "<div class='global-controls'">
    <input class='clickable' type='submit' name='nonce' value='Сохранить'">
  </div>
</form></div>";
  return $render;
}

```

3. Фрагмент коду з файлу `assets/js/main.js`: відповідає за обробку функціональних елементів, а також надання інтерактивності головної сторінки сайту.

```

(function ($) {

  let active_tab, total_price, showB = false;

```

```

function tabActive(tab) {
    $('.ui-tab-item.active').removeClass('active');
    tab.addClass('active');

    active_tab = tab.attr('id');

    $('.ui-parts-list.active').removeClass('active');
    $('.ui-parts-list[data-tab=' + active_tab + ']').addClass('active');
    renderImage();
}

function partActive(part) {
    if (part.hasClass('active')) {
        part.removeClass('active');
    } else {
        $('[data-tab=' + active_tab + '].ui-part-item.active').removeClass('active');
        part.addClass('active');
        renderImage();
    }
}

function itemActive(item, tab) {
    if (tab === false)
        tab = active_tab;
    let part = item.data('part');
    $('[data-tab=' + tab + '].ui-part-item[data-id=' + part + '].ui-item-
item.active').removeClass('active');
    item.addClass('active');
    renderImage();
}

function autoActive() {
    let tabs = $('.ui-tab-item');
    tabActive(tabs.first());
    tabs.each(function () {

```

```

    let id = $(this).attr('id'),
        parts = $('[data-tab=' + id + '] .ui-part-item');
    parts.each(function () {
        itemActive($(this).find('.ui-item-item').first(), id);
    });
});
renderImage();
}

function renderImage() {
    let tab = $('.ui-tab-item.active').attr('id'),
        items = $('.ui-item-item.active[data-tab=' + tab + ']'),
        img_zone = $('.ui-image-zone'),
        price_zone = $('.ui-price-zone'),
        img_str = showB ? 'image_b' : 'image',
        dataplace = $('.ui-popup-wrap input[name=orderdata]'),
        item_list = [];
    total_price = 0;
    img_zone.html("");
    items.each(function () {
        total_price += $(this).data('price');
        img_zone.append('');
        item_list.push({part:$(this).data('part'),item:$(this).data('id')})
    });
    let dataArr = {
        tab:tab,
        items:item_list,
    };
    dataplace.val(btoa(JSON.stringify(dataArr)));
    price_zone.text(total_price + ' ' + price_zone.data('currency'));
}

$(document).ready(function () {

    // set active

```

```
autoActive();

$(document).on('click', '.ui-tab-item', function (e) {
    tabActive($(this));
});

$(document).on('click', '.ui-part-item', function (e) {
    if ($(e.target).closest('.ui-item-item').length > 0) {
        return;
    }

    partActive($(this));
});

$(document).on('click', '.ui-item-item', function (e) {
    itemActive($(this), false);
});

$(document).on('click', '.rotate', function (e) {
    showB = !showB;
    renderImage();
});

$(document).on('click', '.order-me',function (e) {
    $('body').addClass('blur');
    $('.ui-popup-wrap').addClass('open');
});

$(document).on('click', '.ui-popup-wrap .close-popup',function (e) {
    $('body').removeClass('blur');
    $('.ui-popup-wrap').removeClass('open');
});
});

})(jQuery);
```


4. Фрагмент коду з файлу `assets/js/admin.js`: відповідає за обробку функціональних елементів, а також надання інтерактивності адміністративної панелі сайту.

```
(function ($) {
  $(document).ready(function () {
    let template = {
      type:                $('.js-type-template').removeClass('js-type-
template').clone().wrap('<div/>').parent().html(),
      part:                $('.js-part-template').removeClass('js-part-
template').clone().wrap('<div/>').parent().html(),
      item:                $('.js-item-template').removeClass('js-item-
template').clone().wrap('<div/>').parent().html(),
    };
    $('.js-type-template').remove();
    $('.js-part-template').remove();
    $('.js-item-template').remove();
    $(document).on('click', '.clickable.add[data-target]', function () {
      let target = $(this).data('target');
      if (target === 'type') {
        addTab($(this));
      } else if (target === 'part') {
        addPart($(this));
      } else if (target === 'item') {
        addItem($(this));
      }
    });
    $(document).on('click', '.clickable.remove', function () {
      let approved = confirm($(this).text() + '?');
      if (approved) {
        let target = $(this).parent();
        if (target.hasClass('in-tab')) {
          $('.close-type-control span.tab[data-tab=' + target.attr('id') + ']').remove();
        }
      }
    });
  });
});
```

```

    }
    target.fadeOut(300, function () {
        $(this).remove();
        if (target.hasClass('in-tab')) {
            let active = $('.clickable.tab[data-tab=tab_0]');
            if (active.length > 0) {
                swichTab(active);
            }
        }
    });
}
});
$(document).on('click', '.clickable.tab[data-tab]', function () {
    swichTab($(this));
});
function addTab(clickable) {
    let id = 'tab_' + $('form .tabs .group.in-tab').length++;
    $('.close-type-control .clickable.tab.active').removeClass('active');
    clickable.before('<span class="clickable tab active" data-tab="" + id + "">' + id + '</span>');
    $('form .tabs .group.in-tab.active').removeClass('active');
    $('form .tabs').append(template.type.split('theIdReplace').join(id));
}
function addPart(clickable) {
    let id = $('#' + clickable.data('parent') + '.group.in-part').length++;
    clickable.before(template.part.split('theIdReplace').join(clickable.data('parent')).split('theCurrent')
    ).join(id));
}
function addItem(clickable) {
    let id = 'item_' + clickable.closest('.in-part').find('.in-item').length++,
        parent = clickable.data('parent'),
        current = clickable.data('current');
    clickable.before(template.item.split('theIdReplace').join(parent).split('theCurrent').s
    plit('theItemId').join(id));
}

```

```

}
function switchActive(fromElement, toElement, time) {
  if (fromElement.length > 0)
    fromElement.animate({ opacity: 0 }, time, function () {
      fromElement.removeClass('active');
      toElement.css('opacity', 0).addClass('active').animate({ opacity: 1 }, time);
    });
  else
    toElement.css('opacity', 0).addClass('active').animate({ opacity: 1 }, time);
}
function swichTab(switchTo) {
  $('.close-type-control .clickable.tab.active').removeClass('active');
  switchTo.addClass('active');
  let from = $('form .tabs .group.in-tab.active'),
      to = $('# + switchTo.data('tab'));
  console.log(from.length);
  console.log(switchTo.data('tab'));
  switchActive(from, to, 300);
}
$(document).on('click', '.image-label .remove-img', function (e) {
  e.preventDefault();
  let label = $(this).closest('.image-label'),
      fileInput = label.find('input[type=file]').first(),
      hiddenInput = label.find('input[type=hidden]').first(),
      imgSrc = label.find('img').first();
  fileInput.val(null);
  hiddenInput.val("");
  imgSrc.removeClass('active').val("");
});
$(document).on('input change', 'input.action-title', function(){
  let target = $(this).data('target');
  $('.tab[data-tab=' + target + ']').text($(this).val());
});
})(jQuery);

```