

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр на

тему: **«РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ПРОДАЖУ ОДЯГУ З
ВИКОРИСТАННЯМ HTML ТА CSS МОВОЮ JS»**

Виконав: студент 5 курсу, групи ППЗ-51 спеціальності

121___Інженерія програмного забезпечення

(шифр і назва спеціальності)

___Константин В.А

(прізвище та ініціали)

Керівник _____Гуленко В.С.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного
забезпечення
О.В. Негоденко
“ ___ ” _____ 2022 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Константину Віктору Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка web-додатку для продажу одягу з використанням HTML та CSS мовою JS»

Керівник роботи Гуленко В.С., асистент (прізвище, ім'я, по батькові, науковий ступінь, вчене звання) затверджені наказом вищого навчального закладу від “24” лютого 2023 року №26

2. Строк подання студентом роботи “01” червня 2023 року.

3. Вихідні дані до работ:

3.1. Середовище розробки Visual Studio Code

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібнорозробити):

4.1. Дослідження програмних засобів для розробки Web-додатку

4.2. Програмна реалізація Web-додатку

4.3. Приклади використання та тестування Web-додатку

4.4. Висновки

5. Перелік графічного матеріалу.
- 5.1. Титульний слайд
- 5.2. Мета, об'єкт та предмет дослідження
- 5.3. Терміни
- 5.4. Аналіз існуючих рішень
- 5.5. Технічне завдання
- 5.6. Програмні засоби реалізації
- 5.7. Висновки
6. Дата видачі завдання: 25.04.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строки виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.04.-01.05.	виконано
2	Дослідження існуючих інструментів для автоматизації тестування ПЗ	02.05.-04.05.	виконано
3	Проектування архітектури системи	05.05.-10.05	виконано
4	Висновки, оформлення роботи	12.05.-22.05.	виконано
5	Розробка демонстраційних матеріалів	22.05.-25.05.	виконано
6	Попередній захист роботи	26.05.	виконано
7	Здача роботи	09.06	виконано

Студен _____ Константин В.А
 (підпис) (прізвищетаініціали)

Керівник роботи _____ Гуленко В.С.
 (підпис) (прізвищетаініціали)

РЕФЕРАТ

Ключові слова: HTML, CSS, JavaScript, Visual Studio Code.

Об'єкт дослідження – продаж одягу через інтернет.

Предмет дослідження – Web-додаток для продажу одягу.

Мета роботи – підвищити рівень продажу одягу за допомогою Web-додатку з використанням HTML, CSS та JavaScript.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Проаналізувати технічні засоби, що використовуються для розробки та обрати необхідні для створення Web-додатку.
2. Розробити вимоги до Web-додатку на основі аналізу переваг та недоліків існуючих Web-додатків
3. Спроекувати та розробити новий Web-додаток на основі аналізу потреб користувачів.
4. Провести тестування Web-додатку на працездатність.

Практичне значення отримання результатів полягає у створенні Web-додатку для з використанням HTML, CSS та JavaScript.

В роботі розглянуто основні етапи створення Web-додатку і досліджено технічних засобів для розробки Web-додатку.

Галузь використання – Web-додатку може використовувати будь-який користувач в будь якому браузері.

ЗМІСТ

Вступ	9
Як працює Web-додаток:	Error! Bookmark not defined.
1.АНАЛІЗ ТА ВИДІЛЕННЯ ПЕРЕВАГ WEB-ДОДАТКУ	10
1.1 Використання веб-додатків, веб-сайтів та веб-сторінок	10
1.2. Особливості веб-сайтів та веб-додатків	11
1.2.1. Авторизація	11
1.2.2. Інтерактивність	12
1.2.3. Переваги розробки веб-додатків	13
1.2.4. Недоліки розробки веб-додатків	14
1.3. Висновки до розділу	15
2.WEB-ДОДАТОК ПРОДАЖУ ОДЯГУ	Error! Bookmark not defined.
2.1. Обґрунтування вибору інструментів та технологій розробки	16
2.1.1. HTML	16
2.1.2. CSS	17
2.1.3. JavaScript	18
2.1.4. Visual Studio Code	19
2.1.5. SQL	20
2.2. Діаграма прецедентів	20
2.3. Діаграма класів	21
3.ПРОГРАМА WEB-ДОДАТКУ	23
3.1. Розробка інтерфейсу користувача	23
3.2. Висновки до розділу	24

ВИСНОВОК	25
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	26
ДОДАТОК	27

Вступ

Так чи інакше веб-додатки значно змінили наше життя, оскільки вони проникли у всі сфери життя: починаючи від покупки товарів так і до отримання державних послуг онлайн. У нашій статті ми б хотіли докладно зупинитися на тому, що таке веб-додатки, що вони собою представляють. Потім ми торкнемося того, згідно з якими механізмами працюють веб-додатки і які нюанси існують при їх роботі.

Ключові слова: HTML, CSS, JavaScript, Visual Studio Code.

Об'єкт дослідження – продаж одягу через інтернет.

Предмет дослідження – Web-додаток для продажу одягу.

Мета роботи – підвищити рівень продажу одягу за допомогою Web-додатку з використанням HTML, CSS та JavaScript.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Проаналізувати технічні засоби, що використовуються для розробки та обрати необхідні для створення Web-додатку.
2. Розробити вимоги до Web-додатку на основі аналізу переваг та недоліків існуючих Web-додатків
3. Спроекувати та розробити новий Web-додаток на основі аналізу потреб користувачів.
4. Провести тестування Web-додатку на працездатність.

Практичне значення отримання результатів полягає у створенні Web-додатку для з використанням HTML, CSS та JavaScript.

В роботі розглянуто основні етапи створення Web-додатку і досліджено можливості технічних засобів для розробки Web-додатку.

Галузь використання – Web-додатку може використовувати будь-який користувач в будь якому браузері.

1. АНАЛІЗ ТА ВИДІЛЕННЯ ПЕРЕВАГ WEB-ДОДАТКУ

1.1 Використання веб-додатків, веб-сайтів та веб-сторінок

Одним з ключових моментів в розвитку всесвітньої павутини грає веб-розробка - процес створення веб-сайту або веб-додатка. Термін включає розробку додатків електронної комерції, веб-дизайн, програмування на стороні клієнта і серверу, а також конфігурування веб-серверу. Основними етапами веб-розробки є:

- проектування сайту або веб-додатка;
- створення макетів сторінок;
- наповнення;
- обслуговування працюючого сайту або його програмної основи.

Залежно від поточної задачі деякі з етапів можуть бути відсутні, або бути тісно пов'язані один з іншим.

Веб-сторінку можна визначити як окрему сторінку веб-сайту. Доступ до вебсторінки можна отримати за допомогою однієї URL-адреси. Прості веб-сторінки в основному складаються з простої архітектури на основі мови гіпертекстової розмітки HTML і служать лише платформою для відображення інформації. Веб-сторінки можуть містити текст, зображення або музику. Відображення сторінки можна змінити додаванням до неї таблиці стилів CSS або сценаріїв на мові JavaScript. Перегляд вебсторінки не вимагає навігації, на відміну від веб-сайту.

Веб-сайт – це колекція веб-сторінок, з'єднаних між собою, які доступні через мережу, таку як Інтернет і мають єдине доменне ім'я. Він може містити текст, графіку, аудіо, відео, гіперпосилання на інші сторінки тощо. Веб-сайт може бути специфічним для конкретної галузі, конкретним продуктом або конкретними послугами. Вони призначені для ознайомлення відвідувачів з інформацією про їхню галузь, продукти або послуги. Наприклад, звичайний веб-сайт може містити детальну інформацію про серію товарів, але у нього

немає можливості замовнику замовляти товар та здійснювати платежі через веб-сайт. Для цих та інших цілей, пов'язаних з інтерактивом або взаємодією користувача (клієнта) за сервером розробляються вебдодатки.

Веб-додаток - це інтерактивна комп'ютерна програма, розроблена для мережі інтернет, що дозволяє користувачам вводити, отримувати і маніпулювати даними за допомогою взаємодії. Додаток може бути вбудований у веб-сторінку, або сама вебсторінка може бути додатком. Логіка додатка зосереджена на сервері, а браузер найчастіше відповідає лише за відображення інформації, завантаженої з сервера, і за передачу на сервер даних користувача. Веб-додаток отримує запит від клієнта і виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту через мережу з використанням протоколу HTTP. Саме веб-додаток може виступати як клієнт інших служб, наприклад, бази даних або іншого веб-додатку, розташованого на іншому сервері. Яскравим прикладом веб-додатку є система управління вмістом статі Вікіпедії: безліч її учасників можуть приймати участь в створенні мережної енциклопедії, використовуючи для цього браузери своїх операційних систем (будь то Windows, GNU/Linux, IOS або будь-яка інша операційна система) і не завантажуючи додаткових виконуваних модулів для роботи з базою даних статі.

1.2. Особливості веб-сайтів та веб-додатків

1.2.1. Авторизація

Веб-додатки пропонують процес авторизації користувача. Цей процес включає в себе введення призначених для користувача даних для отримання доступу до вебсайту або системі. У деяких випадках це використовується для того, щоб дати більше можливостей, які недоступні неавторизованим користувачам. Таким чином вебдодатки дозволяють своїм відвідувачам обмінюватися миттєвими повідомленнями (чат-платформи, соцмережі, блоги), створюють контент на основі призначених для користувача переваг,

забезпечують необмежений доступ до них, використовують міні-вбудовані програми для розваг.

Процес авторизації важливий для систем, яким потрібно будь-яка особиста інформація про клієнта. Більш того, на цьому етапі, особливу увагу приділяють безпеці. Важливо мінімізувати можливість доступу до особистих даних користувачів стороннім особам.

1.2.2. Інтерактивність

Сайти не пропонують можливості взаємодії з програмою. Його користувачі не мають доступу до розміщення своєї інформації крім заповнення форми для отримання підписки. Найбільш яскравими прикладами типових сайтів є новинні, кулінарні, прогнози погоди.

В той час як веб-додатки створені для взаємодії з користувачами. Веб-додатки теж можуть бути інформативними, але також вони можуть і обробляти інформацію, отриману від користувачів.

Як приклад веб-додатку, розглянемо системи онлайн банкінгу «Ощад24» (рис.1.1). Для того, щоб користуватись основними функціями веб-додатку (наприклад переказ коштів, поповнення мобільного, оплата онлайн послуг та інші), користувачеві потрібно авторизуватись, тобто ввести свій номер телефону та пароль. Або ж створити акаунт.

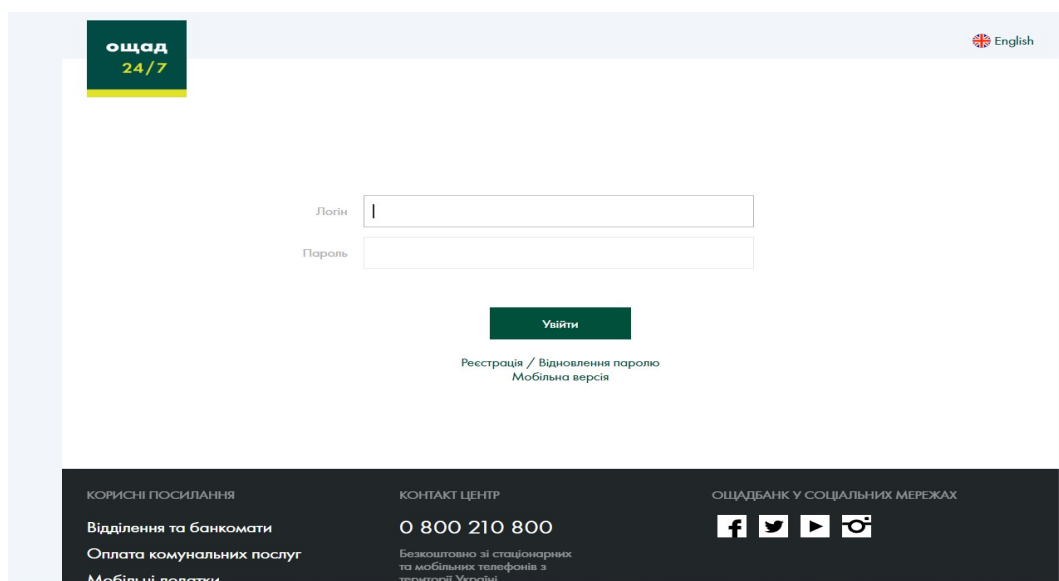


Рис. 1.1. Авторизація у веб-додатку «Ощад24»

1.2.3. Переваги розробки веб-додатків

Незалежність від операційної системи клієнта. Додаток створюється один раз для довільно вибраної платформи і на ній розгортається. Замість того, щоб писати різні версії для MS Windows, Mac OS X, GNU/Linux та інших операційних систем, додаток створюється один раз і відображається на будь-якій платформі.

Відсутність програмного забезпечення для клієнтів. Не потрібно нічого завантажувати і встановлювати. Це спрощує процес користування. Вирішивши запустити свій проект веб додатку (на відміну від десктопного), ви відразу ж зробите його доступним для масової аудиторії.

Доступність. Програма доступна з будь-якої точки планети, де є доступ до Інтернету.

Адаптивність. Можливість користуватися додатком, використовуючи будь-який пристрій (настільний комп'ютер або ноутбук, смартфон або планшет, тощо) з будь-яким розрішенням екрану. Для цього при розробці потрібно використовувати адаптивну верстку. Адаптивна верстка – підхід, що припускає зміну дизайну залежно від поведінки користувача, розміру екрана, платформи і орієнтації девайса. Іншими словами, сторінка повинна автоматично підлаштовуватися під дозвіл, змінювати розмір картинок і так далі. Це дозволить усунути потребу в розробці дизайну для кожного нового пристрою, що з'являється у продажу.

Просте та зручне оновлення. Веб додаток набагато простіше оновлювати, ніж їх настільні аналоги. Щоб оновити веб-додаток, необхідно внести зміни тільки в одному місці: на сервері, на якому обробляється інформація. Оновлення ПО для комп'ютера є набагато більш складним завданням: необхідно розробити, протестувати і, нарешті, поширити оновлення серед усіх кінцевих користувачів. Також суттєво оптимізується процес обслуговування, модернізації інтерфейсу. Всі оновлення активуються автоматично при завантаженні сторінки.

1.2.4. Недоліки розробки веб-додатків

Індивідуальні налаштування браузерів клієнта. Різна реалізація HTML, CSS, DOM і інших специфікацій в браузерах може викликати проблеми при розробці вебдодатків і подальшої підтримки. Крім того, можливість користувача налаштовувати багато параметрів браузера (наприклад, розмір шрифту, кольору, відключення підтримки сценаріїв) може перешкоджати коректній роботі додатку. Проте цю проблему можна вирішити використовуючи так звану кросбраузерну верстку.

Кросбраузерна верстка - це верстка сайту з його однаковою видимістю в різних браузерах. При кросбраузерній верстці потрібно враховувати всі особливості кожного з браузерів, так як користувач заходить з того браузера до якого він звик і ми його не можемо змусити перейти на інший софт, тож повинні підлаштовуватися під нього.

Ослаблений захист. Через відкритий доступ до керування веб-додатком, кількість хакерських атак на такі програми дуже велика. Системи, подібні вебу, RSS і AJAX, схожі тим, що особливих перешкод для їх повторного використання не існує. Велика частина корисного програмного забезпечення має відкриті тексти, а якщо і немає, то існує мало способів захистити свою інтелектуальну власність. Стандартна функція оглядача інтернет "переглянути вихідний код" дозволяє будь-якій людині скопіювати будь-яку веб-сторінку.

Необхідність оптимізації. В успішних веб-проектах на сайт приходять десятки користувачів на секунду. Тому варто замислитися над оптимізацією швидкості виконання програми. Від моменту натискання на посилання до появи результату часто проходить більше 1 секунди (а буває й більше). Тому людина встигає помітити затримку і навіть занудьгувати. Задача програміста - максимально скоротити час реакції додатка.

Недоступність деяких функцій. На відміну від звичайних (нативних) додатків, веб-додаток не може ефективно використовувати всі функції смартфона. Камери, GPS, телефонний набір і інші функції, інтегровані в

пристрій, не завжди добре розроблені для адаптивних веб-сайтів. Також існує велика кількість додатків, які не можуть бути реалізовані в браузері.

Підключення до мережі. Щоб користуватися веб-додатком та його функціями, пристрій користувача обов'язково повинен бути підключений до інтернету.

1.3. Висновки до розділу

На основі проведеного аналізу можна сказати, що на сьогоднішній день розробка веб-сайтів або веб-додатків є однією з основних вимог для створення свого бізнесу, надання послуг, інформування клієнтів, тощо. Майже всі компанії в будь-якій сфері діяльності мають власні веб-сайти/веб-додатки, за допомогою яких їх клієнти можуть з легкістю дізнатися потрібну їм інформацію. В залежності від потреб компанії, вони створюють звичайні веб-сторінки, веб-сайти або веб-додатки. Веб-додатки мають значну перевагу порівняно з веб-сайтами, оскільки користувач може взаємодіяти з інформацією, яка йому представляється. Такі системи не потребують багато часу на розробку, порівняно зі звичайними додатками, і підтримуються на всіх ОС. Розробка веб-додатків зменшує комерційні витрати, оскільки вони потребують менше технічного обслуговування та мають низькі вимоги до системи кінцевого користувача.

Отже, за результатами проведеного аналізу і для розкриття теми дипломного проєкту треба виконати наступне:

- 1) описати загальну структуру додатку за допомогою діаграм;
- 2) розробити алгоритм реєстрації користувача;
- 3) розробити інтерфейс користувача;
- 4) розробити інструкції користувача;
- 5) провести тестування розробленого додатку.

2. WEB-ДОДАТОК ДЛЯ ПРОДАЖУ ОДЯГУ

2.1. Обґрунтування вибору інструментів та технологій розробки

2.1.1. HTML

HTML - це код, який використовується для структурування і відображення вебсторінки та її контенту. HTML не є мовою програмування, це мова розмітки, і використовується, щоб повідомляти вашому браузеру, як відображати вебсторінки, які ви відвідуєте. HTML складається з ряду елементів, які ви використовуєте, щоб вкладати або обертати різні частини контенту, щоб змусити контент відображатися або діяти певним чином. Приклад такого елемента зображено рис. 2.1.

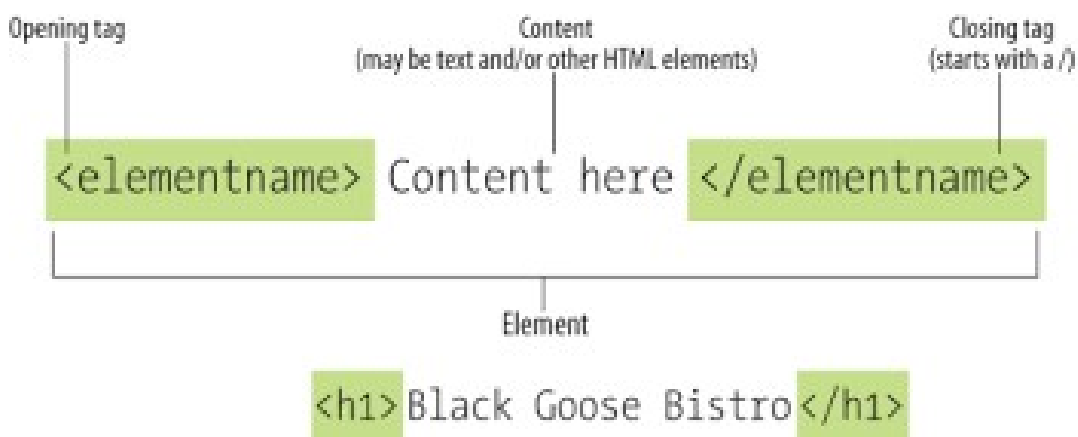


Рис. 2.1. Структура HTML елемента

Основними перевагами використання HTML є:

- 1) підтримка всіма браузерами;
- 2) легко інтегрується з іншими мовами;
- 3) миттєве відображення результату коду.

Тег - це елемент стандартного мови розмітки тексту в Інтернеті — мови HTML. Даними елементами супроводжується початок і закінчення кожної виділеної частини документа. Теги — це одна з кількох властивостей, які користувач може додавати до файлів, щоб їх було легше знаходити та

впорядковувати. Крім тегів, файли мають також інші властивості, такі як дата змінення, ім'я автора та оцінювання. Теги не входять до вмісту файлу. Натомість, теги надають інформацію щодо файлів, на підставі якої користувач може ефективніше шукати та впорядковувати файли. На відміну від інших властивостей, які зазвичай є заздалегідь визначеними, теги можна задавати довільно (наприклад, «фактури», «відпустка» або «особисті документи» тощо). Створені та додані до файлу теги стають однією з його властивостей.

Теги та інші властивості можна використовувати для пошуку та впорядкування файлів. Простіше знайти файл за тегом, який було йому надано, ніж намагатися пригадати, в якій папці цей файл зберігається або його точну назву.

Атрибути тегів, якраз те, що ми вже згадали, – це їх додаткові можливості для форматування тексту. За допомогою тегів браузер визначає, як він повинен відображати ту чи іншу частину тексту або весь текст в цілому: жирним шрифтом, підкресленим, з нового рядка і т.д. Без тегів документ не буде мати ніякого форматування. Для кожного тега визначається безліч можливих атрибутів. Більшість тегів допускають один або декілька атрибутів, проте атрибутів може і зовсім не бути.

2.1.2. CSS

CSS - мова опису зовнішнього вигляду HTML-документа. Це одна з базових технологій в сучасному інтернеті. Практично жоден сайт не обходиться без CSS, тому HTML і CSS діють в єдиній зв'язці. Приклад такого елемента зображено рис. 2.2.

Каскадні таблиці стилів працюють з HTML, але це зовсім інша мова. HTML структурує документ і впорядковує інформацію, а CSS взаємодіє з браузером, щоб надати документу оформлення.

Одна з головних переваг використання CSS - це можливість розділити зміст сторінки від її оформлення. Таке розділення дозволило покращити сприйняття та доступність змісту, забезпечити більшу гнучкість та контроль за

відображенням змісту в різних умовах, зробити зміст більш структурованим та простим, прибрати повторення та ін. Власне це ж і була основна мета створення цієї технології.



Рис. 2.2. Структура CSS елемента

Переваги використання CSS:

- 1) незалежність від платформи;
- 2) прискорення завантаження сторінок і зменшення обсягів інформації;
- 3) можливість розділити зміст сторінки від її оформлення;

Отже, HTML використовується для структурування вмісту сторінки, а CSS - для форматування цього структурованого вмісту. А сукупності HTML та CSS дозволяють творити з документом неймовірні речі і поступово вивчаючи CSS ти почнеш переконуватися в цьому. Інколи створити деякий функціонал, набагато простіше за допомогою CSS аніж на JavaScript.

2.1.3. JavaScript

JavaScript - це динамічна мова програмування, яка застосовується до HTML документу, і може забезпечити динамічну інтерактивність на веб-сайтах. З точки зору веб-розробки, без знань цієї технології неможливо займатися створенням сучасних інтерактивних сайтів. Мова JS - це те, що «оживляє» розмітку сторінок (HTML) і призначений для користувача функціонал сайтів. За допомогою цієї мови реалізується можливість реакції сторінки або окремих її елементів на дії відвідувача. Сьогодні JavaScript є базовою мовою програмування для браузерів. Він повністю сумісний з операційними системами Windows, Linux, Mac OS, а також всіма популярними мобільними

платформами. Всі призначені для користувача дії у вікні браузера створюють події, а програмування на JS дозволяє обробляти їх певним чином.

Стандартний алгоритм роботи виглядає наступним чином:

- 1) користувач виконав певну дію;
- 2) браузер відслідкував подію;
- 3) активується JS-код;
- 4) на сторінці відбувається задана зміна.

Завдання програміста полягає в тому, щоб створити обробники для всіх подій, на які повинен реагувати сайт при взаємодії з користувачем.

2.1.4. Visual Studio Code

Для роботи з файлами потрібен текстовий редактор. Visual Studio Code - безкоштовний і дуже популярний редактор коду від Microsoft. З одного боку він підходить новачкам, тому що його інтерфейс інтуїтивно простий і зрозумілий. З іншого боку в VS Code вбудовані багато можливостей, які цікаві досвідченим розробникам.

При запуску редактора в перший раз відкривається стартова сторінка (рис. 2.2), на якій описані всі можливості редактора. Редактор підтримує безліч мов програмування і легко налаштовується під користувача. Можна встановити різні поєднання клавіш і колірні схеми. Всі посилання інтерактивні і відкривають потрібну частину інтерфейсу або сторінку в браузері.

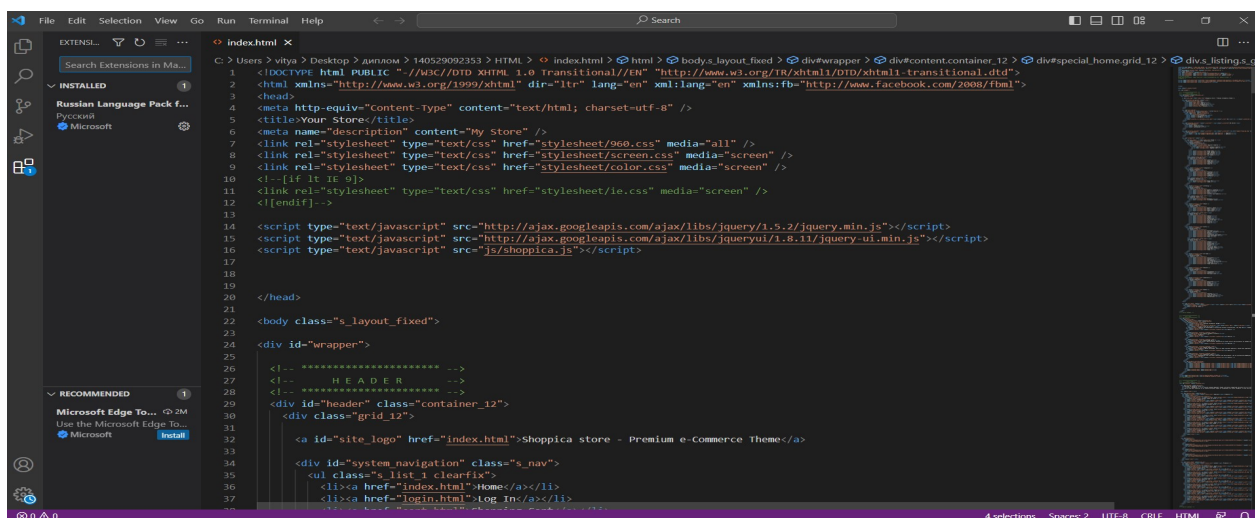


Рис. 2.2. Інтерфейс Visual Studio Code

2.1.5. SQL

Мова SQL є основою багатьох СУБД, тому що відповідає за фізичне структурування й запис даних на диск, а також за читання даних з диска, дозволяє приймати SQL-запити від інших компонентів СУБД і користувальницьких додатків. Таким чином, SQL – потужний інструмент, що забезпечує користувачам, програмам й обчислювальним системам доступ до інформації, що утримується в реляційних базах даних.

Основними перевагами мови SQL є:

- 1) незалежність від конкретних СУБД;
- 2) можливість створення інтерактивних запитів;
- 3) можливість програмного доступу до БД;
- 4) можливість динамічної зміни і структури БД.

2.2. Діаграма прецедентів

Діаграма прецедентів вважається основним видом діаграми візуального моделювання. Вона описує поведінку системи з погляду на неї користувачем. Діаграму прецедентів ще називають діаграмою варіантів використання або use-case діаграмою. Основні елементи діаграми прецедентів є актор, прецедент і відносини між ними.

Актор – це сутність, яка взаємодіє з системою в процесі своєї роботи з нею. Акторами можуть бути люди, зовнішня система, підсистема, клас. Прецеденти – це певні події, що виконуються системою, і ведуть до спостережуваного актором результату. Прецеденти на діаграмі позначаються еліпсами. На рис. 2.3 зображено діаграму прецедентів Web-додатку продажу одягу. На діаграмі зображено три актора, а саме: користувач, зареєстрований користувач та не зареєстрований користувач. Вони зв'язані відношенням узагальнення.

Узагальнення – це відношення між предком та нащадком, з'єднаних стрілкою до предка з незафарбованим трикутником вкінці. В даному випадку

предком є користувач, а нащадками – зареєстрований та не зареєстрований користувачі.

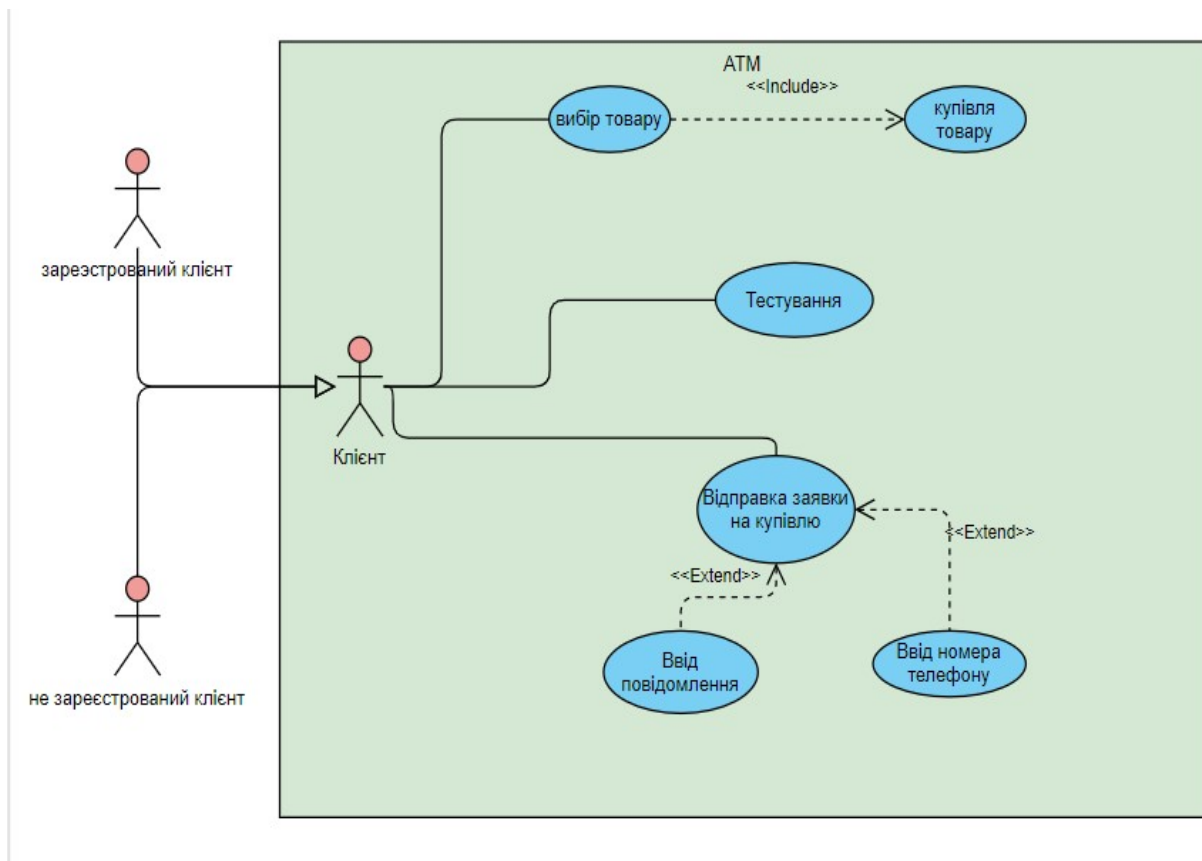


Рис.2.3 Діаграма прецедентів

2.3. Діаграма класів

Діаграма класів – це набір статичних, декларативних елементів моделі. Елементами діаграми класів можуть бути інтерфейси, пакети, відносини, об'єкти та зв'язки.

Клас на діаграмі зображується у вигляді прямокутника, розділеного горизонтальними лініями на три секції. Перша секція містить назву класу. Як правило, ім'я класу складається з одного, максимум двох слів. Друга містить перелік атрибутів класу, які характеризують той чи інший об'єкт цього класу в моделі предметної області. Третя містить перелік операцій, що відображають його поведінку в моделі предметної області.

Атрибут – змістовна характеристика класу, що описує множину значень, які можуть приймати окремі об'єкти цього класу. Операція – це сервіс, що надається кожним екземпляром або об'єктом класу на вимогу своїх клієнтів, в якості яких можуть виступати інші об'єкти, в тому числі і екземпляри даного класу.

Інтерфейс – іменована множина операцій, які характеризують поведінку окремого елемента моделі. Інтерфейс в контексті мови UML є спеціальним випадком класу, у якого є операції, але відсутні атрибути.

На рис. 2.4 зображено діаграму класів Web-додатку продажу одягу. Клас «Головне вікно» є інтерфейсом, тому він не має атрибутів, а містить лише операції. З інтерфейсом зв'язані класи «Одяг», «Повідомлення».

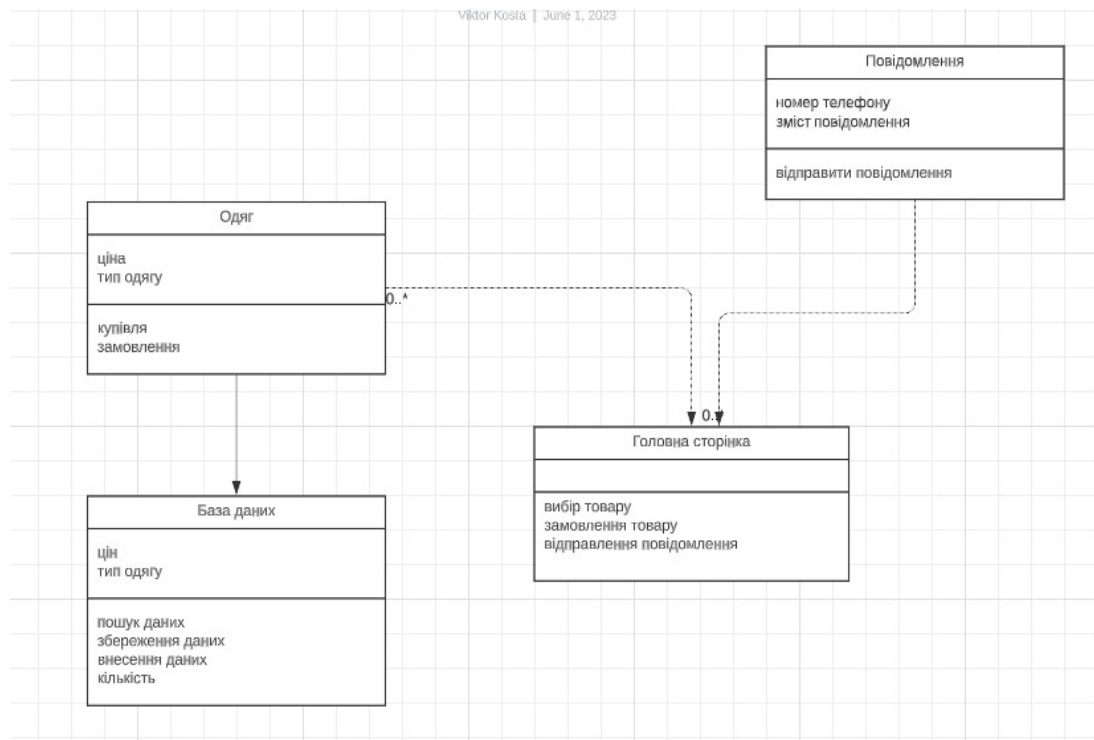


Рис. 2.5 Діаграма класів

3. ПРОГРАМА ВЕБ-ДОДАТКУ

3.1. Розробка інтерфейсу користувача

Інтерфейс веб-додатку – невід’ємна частина та водночас одна з головних, якій слід приділяти максимальну увагу при розробці, оскільки інтерфейс – це те, що бачить та з чим взаємодіє користувач. Інтерфейс веб-додатку розділений на два блоки: статичний і динамічний. Статичний блок інтерфейсу - це модулі, які відображаються на кожній сторінці порталу. Даний блок складається з двох php файлів: header і footer. Header - це навігаційна панель із посиланнями на головну сторінку, сторінку з екскурсіями, сторінку з фактами, контактами, особовим кабінетом та логотипом. Footer містить текст логотипу та рік розробки додатку. Динамічний блок – це контент, кожної з веб-сторінок. На рис. 3.1 зображено головну сторінку веб-додатка.



Рис. 3.1. Головна сторінка веб-додатку

В цьому Web-додатку було використано JavaScript в якому прописано визначати на якому пристрою було відкрито додаток наприклад з телефону чи з персонального комп’ютера. На рис. 3.2 зображено скрипт який допомагає з визначенням пристрою.

```
JS script.js  X
C: > Users > vitya > Desktop > диплом 2 > магазин одягу > JS script.js > ...
1  var isMobile = {
2      Android: function () {
3          return navigator.userAgent.match(/Android/i);
4      },
5      BlackBerry: function () {
6          return navigator.userAgent.match(/BlackBerry/i);
7      },
8      iOS: function () {
9          return navigator.userAgent.match(/iPhone|iPad|iPod/i);
10     },
11     Opera: function () {
12         return navigator.userAgent.match(/Opera Mini/i);
13     },
14     Windows: function () {
15         return navigator.userAgent.match(/IEMobile/i);
16     },
17     any: function () {
18         return (
19             isMobile.Android()
20             || isMobile.BlackBerry()
21             || isMobile.iOS()
22             || isMobile.Opera()
23             || isMobile.Windows()
24         );
25     }
26 };
```

Рис. 3.2. JavaScript код для роботи адаптації пристроїв.

3.2. Висновки до розділу

В цьому розділі був описаний процес розробки інтерфейсу, а саме було показано як реалізовувати адаптивність веб-додатку, адже адаптивність – одна з ключових речей при роботі з додатком користувача.. В додатку створено простий та водночас зручний дизайн. Інтерфейс інтуїтивнозрозумілий, тому користуватися додатком зможе навіть дитина. Також було протестовано роботу адаптивного меню веб-додатку.

ВИСНОВКИ

Робота посвячена розробці Web-додатку інтернет магазину, що підвищить рівень створення Web-додатків в майбутньому.

Проведено аналіз існуючих програмних засобів для розробки Web-додатків. Таким чином розробки було обрано мову програмування JavaScript та середовище розробки Visual Studio Code, також не від'ємною частиною створення Web-додатку є HTML і CSS.

Результат дослідження бакалаврської роботи апробовані на всеукраїнських науково-технічній конференції: Науково технічній конференції: Науково- технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях», м.київ: ДУТ, 5 червня 2023 року.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ

ДЖЕРЕЛ

1. Основи HTML [Електронний ресурс]- <https://guruweba.com/chools.com/;https://dev.to/;https://dev.to/>
2. Поняття CSS [Електронний ресурс]. <https://css.in.ua/>
3. Сайт для питань з програмування [Електронний ресурс]- <https://qna.habr.com/>
4. Сайд для створення діаграм онлайн [Електронний ресурс]- <https://lucid.app/>
5. Основи JavaScript [Електронний ресурс]- <https://ru.stackoverflow.com>
6. Основи HTML та CSS [Електронний ресурс]- <https://guruweba.com/>
7. Готові приклади програми програми [Електронний ресурс]- <https://codepen.io/>
8. Шрифти на CSS [Електронний ресурс]- <https://idg.net.ua/>
9. Робота з картинками HTML і HTML [Електронний ресурс]- <https://d-nik.pro/>
10. Стилзація посилань [Електронний ресурс]- <https://developer.mozilla.org/>
11. Посилання на інформацію що таке ТЕГ [Електронний ресурс]- <https://translations.com.ua/>
12. Що таке HTML [Електронний ресурс]- <https://uk.wikipedia.org/wiki/HTML>
13. Готові скріпти на JavaScript [Електронний ресурс]- <https://xn----7sbbaqhlkm9ah9aiq.net/>

ДОДАТОК

Вся програма Web-додатку магазин одягу

HTML теги

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <nav></nav>
  <div class="content_1">
    <div class="welcom_txt">
      <h1> CLOTHES FOR THE ENTIRE FAMILY</h1>
    </div>
    <div class="icon_1"></div>
    <div class="img_1_1"> </div>
    <div class="img_2_2"> </div>
  </div>
  <div class="men">
    <div class="men_h1">
      <h1>MEN'S CLOTHES</h1>
    </div>
```

```
<div class="img_men_1"> </div>
<div class="txt1">
  <h1 class="txt_inf">045 David Джинси Чоловічі<br>
    Сині Класичного Крою<br>
    C11056207 <br> </h1>
</div>
<div class="men_money1">
<p> 1500 грн</p>
</div>
<div class="img_men_2"> </div>
<div class="txt2"> <h1 class="txt_inf">
  Кросівки Чоловічі<br>
  Let's Shop <br>
</h1></div>
<div class="men_button1">
  <button class="button"><span>Купить</span></button>
</div>
<div class="men_money2">
  <p> 2000 грн</p>
  </div>
  <div class="men_button2">
    <button class="button"> <span>Купить</span>
</button>
  </div>
  <div class="img_men_3"> </div>
  <div class="txt3">
```

```
<h1 class="txt_inf">Сорочка 3 Довгим Рукавом <br>
    Чоловіча Вузького Крою <br> C11053943 <br></h1>
</div>
<div class="men_money3">
    <p> 1600 грн</p>
</div>
    <div class="men_button3">
        <button class="button"><span>Купити</span></button>
</div>
</div>
<div class="women">
    <div class="women_h1"> <h1>WOMEN'S CLOTHES</h1></div>
    <div class="img_women_1">
    </div>
    <div class="women_txt1">
        <h1 class="txt_inf">
            045 David Джинси Жіночі<br>
            Сині Класичного Крою<br>
            C11054994 <br> </h1>
        </div>
        <div class="women_money1">
            <p>1600 грн</p></div>
        <div class="women_botton1">
            <button class="button"><span>Купити</span></button>
        </div>
        <div class="img_women_2">
            </div>
        <div class="women_txt2">
            <h1 class="txt_inf">Сорочка 3 Довгим Рукавом <br>
```

```
                Чоловіча Вузького Крою<br>
                C11053943 <br></h1></div>

        <div class="women_money2">
            <p>2600 грн</p></div>
        <div class="women_botton2">
            <button
class="button"><span>Купити</span></button></div>
            <div class="img_women_3">
                </div>
        <div class="women_txt3">
            <h1 class="txt_inf"> Кросівки Жіночі<br>
                Let's Shop <br></h1></div>
            <div class="women_money3"><p>2000 грн</p></div>
            <div class="women_botton3">
                <button
class="button"><span>Купити</span></button></div>
        </div>
        <script src="script.js"></script>
</body>
</html>
```

CSS СТИЛЬ

```
body {
    margin: 0%;
    overflow-x: hidden;
}

.content_1
{
    width: 100%;
    height: 753px;
    background-size: 100% 100%;
background: linear-gradient(269.75deg, #2B323F 0.26%, #D9D9D9
99.83%);
}

.icon{
    z-index: 1;
    position: absolute;
    left: 30px;
    top: 20px;
    background: url(icon.png);
}

.icon_1{
    z-index: 1;
    position: absolute;
    left: 20px;
    top: 20px;
}
```

```
.img_1_1{
  position: absolute;

  top: 250px;
  left: 153px;
}

.img_1{
  background: url(stock-photo-men-apparel-shirt-jean-
brown.jpg);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 473px;
  height: 434px;
}

.img_2_2{
  position: absolute;
  left: 873px;
  top: 250px;
}

.img_2{
  background: url(img_2.jpg);
background-blend-mode: hard-light;
border: 1px solid rgba(82, 69, 43, 0.15);
border-radius: 53px;
width: 473px;
height: 434px;
}
```



```
.welcom_txt{
width: 20%;
  height: 100px;
  background-size: 100% 100%;
}

h1{
  position: absolute;
  top:15%;
  left: 27%;
  letter-spacing: 1px;
  word-spacing: 1px;
  font-size: 40px;
  font-family:"Lucida Console";
}

.men{
  width: 100%;
  height: 753px;
  background-size: 100% 100%;
  background: linear-gradient(269.75deg, #2B323F 0.26%,
#D9D9D9 99.83%);
}

.men_h1{
  position: absolute;
  width: 20%;
  height: 100px;
  left: 33%;
  background-size: 100% 100%;
```

```
white-space: nowrap;
}
.img_men_1{
  position: absolute;
  left: 200px;
  top: 880px;
}
.img_men1{
  background: url(img_men1.jfif);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 273px;
  height: 334px;
  object-fit: cover;
}
.txt1{
  position: absolute;
  left: 220px;
  top: 1220px;
}
.txt_inf{
  position: absolute;
  top: 15%;
  left: 27%;
  letter-spacing: 1px;
  word-spacing: 1px;
```

```
font-size: 15px;
font-family:"Lucida Console";
white-space: nowrap;
background-size: 100% 100%;
}
.men_money1{
position: absolute;
left: 220px;
top: 1300px;
}
p{
position: absolute;
top:15%;
left: 27%;
letter-spacing: 1px;
word-spacing: 1px;
font-size: 15px;
font-family:"Lucida Console";
white-space: nowrap;
background-size: 100% 100%;
}
.men_button1{
top: 1300px;
position: absolute;
left: 400px;
}
.button{
background: transparent;
border-width: 0;
```

```
position: relative;
cursor: pointer;
font-family: 'Montserrat', sans-serif;
font-size: 14px;
transition: .3s;
}
.button span{
display: block;
position: relative;
z-index: 3;
background: #41294a;
color: white;
padding: 0 30px;
line-height: 40px;
}
.button::before{
content: "";
position: absolute;
width: calc(100% + 5px);
height: calc(100% + 5px);
left: -5px;
top: -5px;
background: #ff2c54;
z-index: 1;
transition: .4s ease-out;
}
.button::after{
content: "";
position: absolute;
```

```
width: calc(100% + 5px);
height: calc(100% + 5px);
right: -5px;
bottom: -5px;
background: #41ebdf;
transition: .4s ease-out .1s;
}
.button:hover::before{
  transform: translate(5px, 5px);
}
.button:hover::after{
  transform: translate(-5px, -5px);
}

.img_men_2{
  position: absolute;
  left: 1000px;
  top: 880px;
}

.img_men2{
  background: url(img_men1.jfif);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 273px;
  height: 334px;
  object-fit: cover;
}

.txt2{
```

```
    position: absolute;
    left: 1020px;
    top: 1220px;
}
.men_money2{
    top: 1300px;
    position: absolute;
    left: 1020px;
}
.men_button2{
    top: 1300px;
    position: absolute;
    left: 1200px;
}
.img_men_3{
    position: absolute;
    left: 600px;
    top: 880px;
}
.img_men3{
    background: url(img_men1.jfif);
    background-blend-mode: hard-light;
    border: 1px solid rgba(82, 69, 43, 0.15);
    border-radius: 53px;
    width: 273px;
    height: 334px;
    object-fit: cover;
}
.txt3{
```

```
    position: absolute;
    left: 620px;
    top: 1220px;
}
.men_money3{
    top: 1300px;
    position: absolute;
    left: 620px;
}
.men_button3{
    top: 1300px;
    position: absolute;
    left: 800px;
}
.women{
    width: 100%;
    height: 753px;
    background-size: 100% 100%;
    background: linear-gradient(269.75deg, #2B323F 0.26%,
#D9D9D9 99.83%);
}
.women_h1{
    position: absolute;
    width: 20%;
    height: 100px;
    left: 33%;
    background-size: 100% 100%;
    white-space: nowrap;
}
```

```
.img_women_1{
  position: absolute;
  left: 200px;
  top: 1650px;
}

.img_women1{
  background: url(img_women1.jpg);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 273px;
  height: 334px;
  object-fit: cover;
}

.women_txt1{
  position: absolute;
  left: 215px;
  top: 1990px;
}

.women_money1{
  position: absolute;
  left: 215px;
  top: 2070px;
}

.women_botton1{
  position: absolute;
  left: 395px;
  top: 2070px;
}
```



```
.img_women_2{
  position: absolute;
  left: 600px;
  top: 1650px;
}

.img_women2{
  background: url(img_women2.jfif);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 273px;
  height: 334px;
  object-fit: cover;
}

.women_txt2{
  position: absolute;
  left: 615px;
  top: 1990px;
}

.women_money2{
  position: absolute;
  left: 615px;
  top: 2070px;
}

.women_botton2{
  position: absolute;
  top: 2070px;
  left: 795px;
}
```

```
.img_women_3{
  position: absolute;
  left: 1000px;
  top: 1650px;
}

.img_women3{
  background: url(img_women2.jfif);
  background-blend-mode: hard-light;
  border: 1px solid rgba(82, 69, 43, 0.15);
  border-radius: 53px;
  width: 273px;
  height: 334px;
  object-fit: cover;
}

.women_txt3{
  position: absolute;
  left: 1015px;
  top: 1990px;
}

.women_money3{
  position: absolute;
  left: 1015px;
  top: 2070px;
}

.women_botton3{
  position: absolute;
  left: 1190px;
  top: 2070px;
}
```

JavaScript коду програми

```
var isMobile = {
  Android: function () {
    return navigator.userAgent.match(/Android/i);
  },
  BlackBerry: function () {
    return navigator.userAgent.match(/BlackBerry/i);
  },
  iOS: function () {
    return navigator.userAgent.match(/iPhone|iPad|iPod/i);
  },
  Opera: function () {
    return navigator.userAgent.match(/Opera Mini/i);
  },
  Windows: function () {
    return navigator.userAgent.match(/IEMobile/i);
  },
  any: function () {
    return (
      isMobile.Android()
      || isMobile.BlackBerry()
      || isMobile.iOS()
      || isMobile.Opera()
      || isMobile.Windows()
    );
  }
}
```

```
};
```