Міністерство освіти і науки України

**Національний університет «Запорізька політехніка»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Комп’ютерні системи і мережі\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(найменування кафедри)

**КУРСОВИЙ ПРОЕКТ**

**(РОБОТА)**

з \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сучасних інтернет технологій\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва дисципліни)

на тему:\_\_Розробка інформаційної системи продуктового магазину\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студента (ки) \_III\_ курсу КНТ-518 групи

Спеціальності 123 комп’ютерна інженерія

освітня програма (спеціалізація) \_\_\_\_Комп’ютерні системи та мережі \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_Костецький Д.В.\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_Касьян К.М.\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_ доцент к.н.т\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали

2020 рік

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Комп’ютерні системи і мережі\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дисципліна \_\_\_\_\_\_\_\_\_\_\_\_**«**Організація баз даних»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність \_\_\_\_\_\_\_\_\_\_\_123Комп’ютерна інженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс 2 Група КНТ-518 Семестр 2

**ЗАВДАННЯ**

**на курсовий проект (роботу) студентові**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Костецького Дмитра Віталійовича **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи): \_\_Розробка інформаційної системи\_\_\_\_\_\_\_\_\_\_\_\_\_

Генератора особистостей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи):\_\_\_17.05.2020\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до проекту (роботи): опис предметної області\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. ЗМІСТ розрахунково-пояснювальної записки (перелік питань, що їх належить розробити): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 аналіз предметної області;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 проектування структури та алгоритму роботи Web-системи;\_\_\_\_\_\_\_\_\_\_\_\_\_

3 Проектування бази даних;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4 Розробка програмного коду\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Розгортання та тестування системи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень):

1 Схема структурна розроблюваної WEB-системи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Дата видачі завдання:\_20.05.2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + - 1. **КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсового  проекту (роботи) | Строк виконання етапів курсового проекту ( роботи ) | * + 1. Примітка |
| 1 | Отримання завдання на курсовий проект | **01.11.2020** |  |
| 2 | Опис та аналіз предметної області | **05.11.2020** |  |
| 3 | Аналіз технічного завдання | **10.11.2020** |  |
| 4 | Проектування структури та алгоритму роботи розроблюваної Web-системи | **20.11.2020** |  |
| 5 | Розробка компонентів Webсистеми | **25.11.2020** |  |
| 6 | Розгортання та тестування розробленої Web-системи | **01.12.2020** |  |
| 7 | Складання пояснювальної записки | **05.12.2020** |  |
| 8 | Виконання графічної частини |  |  |
| 9 | Представлення закінченої роботи на перевірку | **15.12.2020** |  |
| 10 | Захист курсової роботи | **20.12.2020** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Костецький Д. В.\_\_­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( підпис ) (прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Касьян К.М.\_\_\_\_\_\_\_\_\_\_

( підпис ) (прізвище та ініціали)

«17» Листопада 2020 р.

**РЕФЕРАТ**

Изм.

Лист

№ документа

Підпис

Дата

Лист

1

**13.02070849.51809 ПЗ**

Розроб.

Костецький Д.В.

Преревір.

Т. Контр.

Н. Контр.

Утв.

Розробка інформаційної системи

продуктового магазину

Лит.

Листів

НУ«Запорізька Політехніка»

Реценз.

Масса

Масштаб

ПЗ: 26 с., 20 рис., 10 табл., 8 джерел.

СКБД, HTML, CSS, JS, PHP, SEO.

Об’єкт розробки- база даних продуктового магазину

Мета проекту- розробка бази для автоматизації обліку товару у продуктовому магазині.

В ході розробки було проаналізовано предметну область. Проведено концептуальне проектування, в результаті якого були визначені сутності бази даних, зв’язки між ними і побудована ER-модель. Приведено аналіз сучасних СКБД і для розробки бази даних була СКБД MS Access. Проведено логічне і фізичне проектування.

Розроблена база даних містить форми для заповнення всіх таблиць, форму для надходження товарів на склад, форми для їх продажу, звіти для перегляду даних про надходження і видачу товарів, має зручний та зрозумілий інтерфейс.

ЗМІСТ

Вступ…………………………………………………………………………….6

Висновок…………………………………………………………………...…....

Перелік джерел посилань………………………………………………...…….

Додаток А……………………………………………………………………….

## Вступ

Веб-технології ознаменували собою масовий прорив Інтернет ресурсів до широкого кола користувачів, і активне їх поширення представляється необоротним процесом.

Більшість вчених і фахівців вважають, що ХХI століття - століття інформації та інформаційних технологій. Економічний розвиток світового співтовариства обумовлено двома незворотними поступальними процесами: швидким розвитком інформаційних технологій і жорсткою конкуренцією на ринку інтелектуальних послуг. Провідна роль Інтернет в цих процесах сьогодні стає очевидною.

Актуальність теми дослідження: полягає в тому, що на сьогоднішній день створюється безліч різних сайтів і додатків в яких розробники використовують бази даних для зберігання даних при реєстрації користувачів, і робота тестувальників перевіряти на те як будуть відображатися дані, що вводяться користувачами і як в подальшому вони будуть виглядати в продукті, завдяки чому вони мають потребу в сайтах, які будуть автоматично генерувати ці дані і прискорювати цей процес, але в основному ці сайти генерують дані російською мовою, за цим розробка сайту для генерації особистості на українській мові є актуально і затребуваною.

Цільова аудиторія: розробники, тестувальники і люди які не хочуть вводити реальні дані на сайтах при реєстрації.

Метою курсової роботи є формування теоретичних знань з проектування web-сайту і практичних навичок з розробки законопроекту за допомогою сучасних технологій.

Результатом має вийти сайт для генерації особистості, який полегшить, а також прискорить роботу розробників і тестувальників.

Для виконання курсового проекту були поставлені наступні завдання: познайомитися з теоретичними поняттями розробки сайтів, проаналізувати роботу декількох сайтів для генерації особистості, вибрати програмні засоби для реалізації і розміщення сайту в мережі Інтернет, вивчити засоби просування сайтів для подальшого використання.

**1 Теоретична частина**

**1.1 Аналіз предметної області**

Створення сайту потрібно починати з аналізу за деякими причинами:

по-перше, зрозуміти, для яких цілей створюється сайт;

до-друге, потрібно знати яку інформацію буде на ньому розміщено;

по-третє, необхідно стиль та кольорову гаму для дизайну

Необхідність розробки даної системи дозволить скоротити час розробникам та тестувальникам який витрачається на вигадування даних та запису їх у відповідні поля, або для звичайних людям які не мають високих навичок для роботи з комп’ютером, та не бажають вводити свої справжні данні на підозрілих сайтах.

Про аналізувавши велику кількість аналогічних сайтів за допомогою пошукової системі google я зустрічав сайти які генерують не так багато різних критеріїв для створення особистості, деякі генерують лише ПІБ та пошту, деякі розширюють цей список додаючи до нього ще логін і пароль, та однією з най головніших проблем було те що немає сайтів які створювали б цю інформацію українською мовою, всі сайти з генерацією в основному генерують данні російською мовою, тому було прийнято рішення створити сайт на якому будуть генеруватись максимально багато різних даних українською мовою. Переглядаючи багато різних сайтів я зрозумів що дизайн повинен бути простим, основна інформаційна частина розташовується у центрі екрану, переважно вони виконуються у стилі мінімалізму, який складається з мінімальної трансформації об’єктів, простоти та однаковості форм, акуратності та точності до маленьких дрібниць .

В зв’язку з написаним вище було прийняте рішення, що для генерування даних потрібно використовувати такі поля: ПІБ, адреса проживання, професія за якою працює особистість, мобільний номер телефону, електрона пошта, логін, пароль, іноземна мова яку вивчає людина, дата народження, улюблений колір, зріст та вагу, так як це основні критерії які описують сутність особи.

Основне завданням з функціонального боку є збирання бази даних сайту з великою кількістю даних для генерації. Це пов’язано з ти що записи не повинні повторюватись, та повинні буде схожими на справжні, а не просто згенерованим набором символів, тому потрібно зібрати наступні бази: імен, прізвищ, по-батькові, професій, та адрес, інші значення можна генерувати за допомогою мови програмування, так як ті данні складаються з певного набору символів, і головною перевагою цього вибору буде те що під час генерації за допомогою цієї мови затримка для генерування буде майже дорівнювати нулю, так як генерація виконуватиметься на її комп’ютері.

Виходячи з вище написаного найголовнішою вимогою до сайту є його якість оформлення та простота, так як ним будуть користуватись як звичайні люди так і ті хто мають високий рівень навичок при роботі з комп’ютером.

Для звичайних людей сайт повинен вміти генерувати або всі данні одразу або за натискання певної кнопки обране поле. Дизайн повинен бути зручним та красивим. На кожній сторінці повинна бути кнопка, після натискання якої буде відображена коротка інструкція в якій буде написано, як працювати з сайтом. Також повинна бути присутня кнопка яка буде копіювати згенероване поле, щоб користувач не витрачав час на виділення тексту та його копію, це за нього зробить одне натискання на цю кнопку, та поле для вибору статі генерованої особи.

Для розробників і тестувальників повинно бути розроблено ще дві додаткові сторінки, які будуть по своєму генерувати інформацію:

Перша буде створювати інформацію аналогічно тій що генерує для звичайних людей, за відмінністю того що в ній можна буде за допомогою галочка обирати ті поля які потрібно згенерувати, та поля в якому можна вказати цих генеруючих даних. Також повинна бути додана кнопка для вибору всіх полів одним натисканням.

Друга буде створювати SQL запит за допомогою якого можна створити базу даних та занести до неї данні. У формі буде багато полів, які будуть відповідати назві певного атрибута. Тобто най першим буде поле з назвою бази даних. Інші поля працюють за іншим принципом, щоб критерій був доданий до цього запиту під час генерації, йому потрібно привласнити ім’я, і з значенням всіх обраних полів поля буде створена таблиця, яка буде в яку буде складатися з цих атрибутів. Також будуть присутні кнопки: для генерації даних, очистки форми від введених назв, та кнопка яка буде скорочувати час копіювання всього отриманого запиту а саме для його завантаження одразу у файли з розширенням .sql.

І останньою буде сторінка з питаннями, які можуть виникнути у користувачів під час експлуатації сайту, і вона буде зберігати короткі та змістовні відповіді на них.

Навігація на сайті повинна бути зведена до мінімуму зайвих переходів і реалізована у вигляді меню, яке знаходитиметься у горі сайту, та буде дублюватись на кожній його сторінці.

**1.2 Мова розмітки гіпертекстових сторінок HTML**

HTML (HyperText Markup Language - «мова гіпертекстової розмітки») - визначає зміст і структуру веб-контенту. Інші технології, крім HTML, зазвичай використовуються для опису зовнішнього вигляду / уявлення (CSS), функціональності / поведінки (JavaScript) веб-сторінки, або обробки даних на сервері (PHP) але про них трішки пізніше.

Під гіпертекстом розуміються посилання за допомогою, яких з'єднують веб-сторінки один з одним або в межах одного веб-сайту, або між веб-сайтами. Посилання є фундаментальним аспектом Інтернету.

Можна писати сайти без знання мови HTML, оскільки тексти HTML можуть створюватися різними спеціальними редакторами і конвертерами. Але вони пишуть не оптимізований код та додають на потрібні теги, через це вони обмежені в своїх можливостях, містять помилки або проводять поганий HTML код, який не працює на різних платформах.

HTML-елемент відрізняється від простого тексту в документі за допомогою "тегів", які складаються з імені елемента оточеного знаками "<" і ">". Ім'я елемента всередині тегу не чутливе до регістру тому його можна написати як в верхньому так і в нижньому регістрі, або змішано.

Прийнято що HTMLдокумент повинен мати таку форму:

<html>

<head>

<meta>

<title>...</title>

</head>

<body>

</body>

</html>

HTML документ складається з :

1. Оголошення документу HTML

<HTML> і </ HTML>. Пара цих тегів повідомляє програмі перегляду (браузеру) що між ними укладено документ у форматі HTML, причому першим тегом в документі повинен бути тег <HTML>, а останнім - </ HTML>. Даний елемент є самим зовнішнім, так як між його початковим і ліжковим тегами повинна знаходиться вся WEB - сторінка. Все, що знаходиться за межами тегів, не сприймається браузером як код HTML і ніяк їм не обробляється.

2. Технічна інформація про сторінку

Розділ <head> ... </ head> складається з: заголовок, опис, ключові слова для пошукових машин, кодування і т.д. Введена в ньому інформація не відображається у вікні браузера, однак містить дані, які вказують браузеру, як слід обробляти сторінку.

Обов'язковим тегом розділу <head> являється тег <title>. За допомогою нього можна встановити ім’я сторінки яке буде відображатись у заголовку вгорі. Він повинен бути коротким, щоб повністю поміститися в заголовку. Текст заголовка повинен містити максимально повний опис вмісту веб-сторінки.

Необов'язковим тегом розділу <head> є одинарний тег <meta>. З його допомогою можна задати опис вмісту сторінки і ключові слова для пошукових машин, автора HTML-документа і інші властивості метаданих. Елемент <head> може містити кілька елементів <meta>, тому що в залежності від використовуваних атрибутів вони несуть різну інформацію.

Також може використовуватись тег <style>, за допомогою нього можна задати стилі, які використовуються на сторінці. Для завдання стилів в HTML-документі використовується мова CSS, про яку поговоримо пізніше. Таких елементів на сторінці може бути кілька.

За допомогою тега <link> можна покращити читання коду тим, шо код написаний на мові css можна винести до окремого файлу, та підключати його до тих сторінок, де це необхідно.

Тег <script> дозволяє приєднувати до документа різні сценарії. Закриває тег обов'язковий, при цьому текст сценарію може розташовуватися або всередині цього елемента, або в зовнішньому файлі.

3. Тіло сторінки

Головною частиною документа є його тіло. Воно йде відразу за заголовком і знаходиться між тегами <BODY> і </ BODY>. Перший з них повинен стояти відразу після тега </ HEAD>, а другий - перед тегом </ HTML>. Тіло HTML-документа - це місце, куди автор поміщає інформацію, відформатовану засобами HTML.

Основні теги які використовуються у тілі сайту:

<B> Задає напівжирний шрифт уривка тексту, не надаючи акцент або на важливості виділеного.

<Br> Перенесення тексту на новий рядок.

<Div> Тег-контейнер для розділів HTML-документа. Використовується для угруповання блокових елементів з метою форматування стилями.

<Form> Форма для збору і відправки на сервер інформації від користувачів. Чи не працює без атрибута action. Корисна для отвравкі даних на сервер і обробки за допомогою php.

<H1-h6> Створюють заголовки шести рівнів для пов'язаних з ними розділів.

<Header> Секція для вступної інформації сайту або групи навігаційних посилань. Може містити один або кілька заголовків, логотип, інформацію про автора.

<Hr> Горизонтальна лінія для тематичного поділу параграфів.

<Input> Створює багатофункціональні поля форми, в які користувач може вводити дані. Вони мають дуже много разних типів.

<P> Частини в тексті.

<Span> Контейнер для малих елементів. Можна використовувати для форматування уривків тексту, наприклад, виділення кольором окремих слів.

<Table> Тег для створення таблиці.

<Ul> Створює маркований список.

**1.3 Основні технології**

Frontend (це те що виконується у браузері) технології:

frontend – это код, который скачивается в браузере, и который видят пользователи. В этом коде описаны все элементы, цвет стрелочки, цвет текста, верстка, что в этом месте должен быть именно такой отступ. То есть все взаимодействия внутри браузера – это frontend.

При написанні сайту на чистій мові програмування можна зробити все, що завгодно, і обмежують нас тільки можливості самої мови, при написанні за допомогою фреймворку з одного боку, допомагає і прискорює розробку, а з іншого, накладає певні обмеження, ще одним варіантом для розробки є CMS, який є конструктором, в якому по частинах збирається потрібний проект. Його скоріше не програмують, а налаштовують, обмежень в ньому дуже величезна кількість, вийти за межі коробки складно і неефективно.

Існує багато технології, із основних це HTML, CSS та JavaScript, або фреймворки які написані на мові JS, які спрощують життя розробникам AngularJS, React.js, Vue.js

CSS

CSS – це мова таблиць стилів, яка дозволяє прикріплювати стиль (наприклад, шрифти і колір) до структурованих документів HTML. Зазвичай CSS-стилі використовуються для створення і зміни стилю елементів веб-сторінок. Стилі написані на цій мові можна зберігати як у документі HTML у тегу <style></style>, так і в зовнішньому файлі підключивши його за допомогою тегу <link>. Відокремлюючи стиль подання документів від вмісту документів, спрощується читання коду.

CSS підтримує таблиці стилів для конкретних носіїв, тому розробники можуть адаптувати подання своїх документів як до візуальних браузерів, так і до кишенькових пристроїв.

Каскадні таблиці стилів описують правила форматування елементів за допомогою властивостей і допустимих значень цих властивостей. Для кожного елемента можна використовувати обмежений набір властивостей, інші властивості не будуть чинити на нього ніякого впливу.

Оголошення стилю складається з двох частин: селектора і оголошення. В HTML імена елементів нечутливі до регістру, тому «h1» працює так само, як і «H1». Оголошення складається з двох частин: ім'я властивості (наприклад, color) і значення властивості через двокрапку (grey). Селектор повідомляє браузеру, який саме елемент форматувати, а в блоці оголошення (код в фігурних дужках) перераховуються команди форматування - властивості і їх значення.

JavaScript

JavaScript - один з найпопулярніших мов програмування в світі. Це також один з трьох головних мов програмування для веб-розробників. Він як і css може бути розміщений в самому документі HTML

або винесено в окремий файл.

До головних плюсів JS можна віднести те що він легкий у вивченні і його можна використовувати в самих різних варіантах застосування, наприклад в поліпшенні функціональності сайту, зробити їх «живими».

Ще не мало важливим плюсом є те що JavaScript виконується в клієнтському браузері і обробляє команди на комп'ютері кінцевого користувача, і не навантажує цим сервера, що призводить до зниження навантаження на сервер і збільшення швидкості роботи програми так як код виконується відразу в браузері і не витрачається час для його відправлення та прийому з сервера. JavaScript був розроблений компанією Netscape і навряд чи є сайти, які не використовують його.

**Angular.JS**

Angular.JS має досить низький поріг входження в порівнянні з багатьма подібними рішеннями. Документація носить злегка суперечливий характер. Вона ніби добре структурована, є приклади коду, але деякі речі висвітлені вкрай слабо. З ними доведеться розбиратися самостійно шляхом вивчення початкових кодів. Розробники Angular.JS відійшли від традиційної ідеї: «HTML ворог і потрібно з ним боротися». Замість цього вони вирішили природним чином розширити мову розмітки, шляхом введення додаткових директив.

Директиви є однією з ключових можливостей Angular.JS. Вони дозволяють розробнику описати поведінку окремих елементів і розширити синтаксис HTML. До складу angular входить лише базовий набір директив. Однак, його можна розширити своїми власними напрацюваннями. При правильному їх налаштуванні, вони можуть використовуватися і в інших проектах.

Минусами Angular.JS является то что в его основе лежит сложный язык программирования и пользователи сталкиваются с ошибками во время миграции между версиями.

**React**

React — это инструмент для создания пользовательских интерфейсов. Его главная задача — обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты. Каждый выделенный фрагмент страницы, считается компонентом.

Компонент React — это, если по-простому, участок кода, который представляет часть веб-страницы. Каждый компонент — это JavaScript-функция, которая возвращает кусок кода, представляющего фрагмент страницы. Для формирования страницы мы вызываем эти функции в определённом порядке, собираем вместе результаты вызовов и показываем их пользователю. Все компоненты, маленькие или большие, могут использоваться повторно, даже в разных проектах.

Минусами React является неупорядоченность документации, большой выбор инструментов из-за которых пользователи часто становятся в ступор и для освоения всех его нюансов потребуется очень много времени.

Backend (працює на сервері обролює данні або відправляє результат до браузеру користувача):

Основні мови програмування які використовуються Ruby, PHP, Python та фреймворки **Django Node.js**

PHP

Ruby

Python

Node.js

Django

Django - это веб-фреймворк Python. Он может использоваться в основном с любым фронтенд фреймворком (включая описанные выше). Он также является хорошим решением для любых типов веб-сайтов из-за множества доступных сторонних пакетов. С ростом популярности Python стоит рассмотреть Django для серверной части вашего веб-приложения.

Для реалізації бази даних буда обрана mySQL

Елемент (element) - конструкція мови HTML.Ето контейнер, що містить дані і дозволяє відформатувати їх певним чином. Будь-яка Web-сторінка являє собою набір елементів. Одна з основних ідей гіпертексту - можливість вкладення елементів.

Тег (tag) - початковий або кінцевий маркери елемента. Теги визначають межі дії елементів і відокремлюють елементи один від одного. У тексті Web- сторінки теги полягають в кутові дужки, а кінцевий тег завжди забезпечується косою рисою.

Атрибут (attribute) - параметр або властивість елемента. Це змінна, яка має стандартне ім'я і якої може присвоюватися певний набір значень: стандартних або довільних. Передбачається, що символьні значення атрибутів полягають в лапки, але деякі браузери дозволяють не використовувати лапки. Це пояснюється тим, що тип атрибута завжди відомий заздалегідь. Атрибути розташовуються всередині початкового тега і відокремлюються один від одного пробілами.

Гіперпосилання - фрагмент тексту, який є покажчиком на інший файл або об'єкт. Гіперпосилання необхідні для того, щоб забезпечити можливість переходу від одного документа до іншого.

Технічне завдання на розробку Web-додатку

## Обоснование выбора технологий разработки сайта

На первом этапе проектирования web-сайта необходимо выполнить анализ предметной области, т.е. определить объекты предметной области и связи между объектами.

Была выбрана те

2. Проектирование структуры (состава и связей) Вашей системы. Проектирование алгоритма работы системы. Проектирование остальных компонентов системы, например, БД, интерфейс и т.д.

## 4. Алгоритм работы

Установка програмного забезпечення.

Що б була можливість розроблятися сайт на комп'ютері слід встановити програмне забезпечення під назвою «OpenServer», воно емулює роботу сервера на звичайному комп'ютері, але на такий сервер можуть заходити тільки ті пристрої які підключені до локальної мережі

Це програмне забезпечення включає в себе:

- Web-сервер Apache - програма, призначена для створення основи локального сервера;

- СУБД MySQL-система для управління базами даних;

- Інтерпретатор скриптів PHP - необхідна для розробки і використання на сервері розширень і додатків, написаних на мові програмування PHP;

- Утиліта phpMyAdmin -зручний інтерфейс для адміністрування СКБД MySQL.

Воно призначене для локальної роботи з сайтом, не виходячи в Інтернет. Включає в себе безліч популярних серверів сторонніх виробників, що працюють в Windows. З його допомогою можна легко тестувати сайти в, а після завершення тестувань просто перенести їх на реальний хостинг.

3. Программная реализация и тестирование Вашей системы (валидация кода, адаптивность/ кроссбраузерность, СЕО). Исследование ее характеристик и демонстрация того, что та эффективность, которая ожидалась от системы, достигнута.

## Дослідна експлуатація та впровадження проекту

Опис етапів розміщення в мережі

Для опублікування в мережі інтернет будь-якого веб-сайту необхідно дві складові: домен і сервер, домен - унікальна адреса, за якою можна звертатися до сайту, і сервер (хостинг) - місце зберігання всіх файлів для коректної роботи сайту і обрабки серверних скриптів. Хостинг - це, в своєму роді, той же комп'ютер, тільки доступний в мережі 24 години на добу, 7 днів на тиждень.

Для публікації сайту в інтернет був обраний сайт sprinthost.ru, так як на цьому сайті є 30 безкоштовного користування, а цього достатньо щоб в повній мірі зрозуміти як все працює, бо я використовую цей ресурс тільки в навчальних цілях, і немає необхідності реєструвати хостинг на платній основі, ще одним із плюсів стало те що сервер зберігає данні на дуже швидких ssd дисках типу nvme, тому робота з базою даних та виконання скриптів буде швидше ніж на хостингах які зберігають данні на жорстких дисках, також цей хостин дає безкоштовне доменне ім’я та підтримує всі технології які я використовую на своєму сайті.

Зареєструвавшись на сайті ми отримуємо доступ до панелі керування, за допомогою якої завантажуємо на хостинг ввесь код отриманого сайту від час розробки, та саму базу даних в якій зберігаються дані для генерації полів.

Верстка, програмування та тестування

**4.1 Оптимизация кода**

Очень часто, заглядывая в исходный код страницы можно наблюдать самые невероятные заборы из кусков совсем ненужного кода. Появляться там он может по двум причинам. Первая – это когда сайт делается с помощью визуальных средств разработки. Эта причина маловероятна, если сайт делали профессиональные сайтостроители, которые, как правило, достаточно хорошо знают HTML, чтобы сверстать страничку вручную, или поправить её код после визуального редактора.

Вторая причина заключается в том, что даже профессиональные верстальщики пока далеко не все уделяют должное внимание юзабилити, или попросту говоря – удобству использования своего творения в будущем. Это делается, конечно, не со злым умыслом, но факт остаётся фактом.

В качестве непонятных наворотов могут встречаться неоптимальные решения в вёрстке, большие фрагменты кода JavaScript, прописанные в коде стили элементов, брошенные обрывки кода и др.

Любой специалист по юзабилити скажет, что такое отношение к вёрстке может быть чревато долгой загрузкой страницы, что вызывает раздражение у посетителей сайта и создаёт у них неприятное впечатление от посещения сайта.

Для того, чтобы все страницы сайта были быстрее проиндексированы поисковой системой код страницы желательно иметь легче. Кроме того, у некоторых поисковых систем существует ограничение на размер кода индексируемого документа. У всех поисковых систем этот объём может быть разным – 100 – 200 Кб.

Как более дружественную для поисковых систем вёрстку, можно назвать блочную. У блочной вёрстки существуют свои незначительные недостатки, но они с лихвой окупаются более меньшим объёмом кода по сравнению с табличной. Такие странички быстрее загружаются в браузер, так как имеют объём кода на 30–50% меньше, чем у табличной.

**Висновок**

У курсовій роботі були розглянуті актуальні технології розробки сучасного Web-сайту такі як: HTML, CSS, JavaScript і PHP які створені для розробки сайтів, основними варіантами представлення інформації на сайті, а так само використання програм застосовуються для розробки і створення Web-сайту на комп'ютері;

Після аналізу теоретичної частини була розроблена структура і дизайн веб-сайту, який інтуїтивно зрозумілий, простий у використанні і відповідає всім вимогам.

Останнім етапом розробки Web-сайту було додавання інструкції користування для кожної сторінки і створення окремого пункту меню пі переході за яким відкриваються часто задаються.

В результаті проведених робіт на базі обраних сучасних технологій був створений сучасний Web-сайт. Який має інтуїтивно зрозумілий інтерфейс, оформлений в приємній колірній гамі і адаптований під екрани безлічі пристроїв

**Перелік джерел посилань**

1. Оліщук А. В. Розробка Web-додатків. Професійна робота. - М .: «Вільямс», 2006. - С. 352. - ISBN 5-8459-0944-9.

2. Віктор Ромашов. CMS Drupal: Система керування вмістом сайту. - Пітер, 2010. - 255 p. - ISBN 978-5-49807-241-8.

3. Хаген Граф. Створення веб-сайтів. - Видавничий дім «Вільямс», 2009. - 312 p. - ISBN 978-5-8459-1506-1.

4. Фрімен Ерік, Фрімен Елізабет. Вивчаємо HTML, XHTML і CSS = Head First HTML with CSS & XHTML. - П .: «Пітер», 2010. - 656 с. - ISBN 978-5-49807-113-8.

5. Стівен Шафер. HTML, XHTML і CSS. Біблія користувача, 5-е видання = HTML, XHTML, and CSS Bible, 5th Edition. - М .: «Діалектика», 2010. - 656 с. - ISBN 978-5-8459-1676-1.

6. Девід Сойєр Макфарланд. Нова велика книга CSS = CSS: The Missing Manual. - Санкт-Петербург: Пітер, 2017.-720 с.- 1000 екз.- ISBN 978-5-496-02080-0.

7. Дакетт, Джон. Javascript і jQuery. Інтерактивна веб-розробка. - М., 2017. - 640 с. - ISBN 978-5-699-80285-2.

8. Зандстра М. PHP. Об'єкти, шаблони і методики програмування. - 4-е изд .. - СПб .: «Діалектика», 2015. - С. 576. - ISBN 978-5-8459-1922-9.

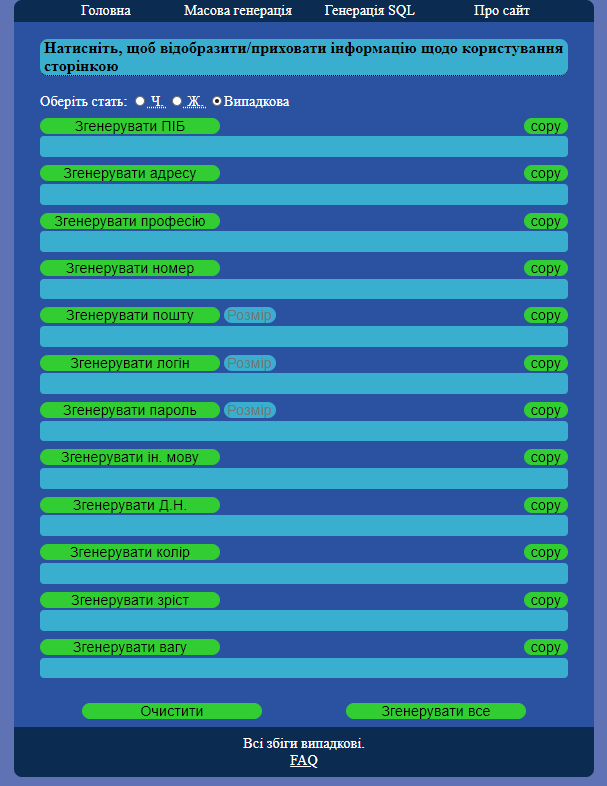
9. Кузнецов Максим, Сімдянов Ігор. Об'єктно-орієнтоване програмування на PHP. - СПб .: «БХВ-Петербург», 2007. - С. 608. - ISBN 978-5-9775-0142-2.

10. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5. 4-е изд. - СПб .. - М .: «Діалектика», 2016. - 768 с. - ISBN 978-5-496-02146-3.

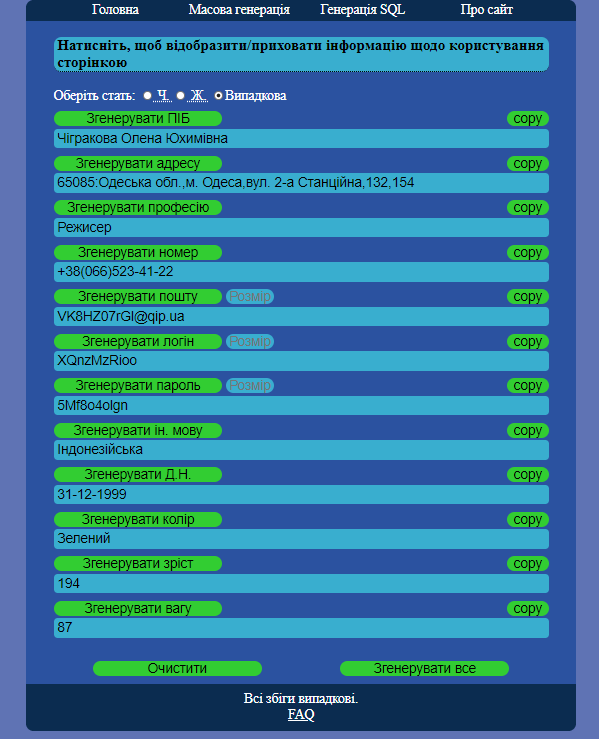
11. Ашманов І. С., Іванов А. А. Просування сайту в пошукових системах. - М .: «Вільямс», 2007. - 304 с. - ISBN 978-5-8459-1155-1.

12. Осипенков Я.О. Google Analytics для googлят: Практическое руководство по веб-аналитике. — 2018. — 580 с.

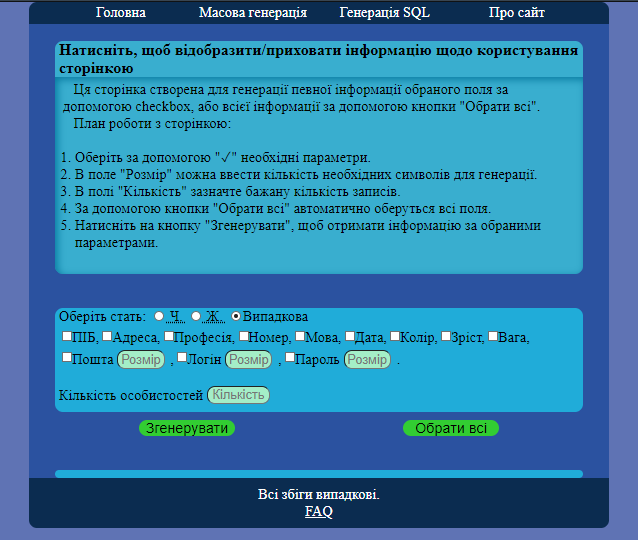
Додаток А.1



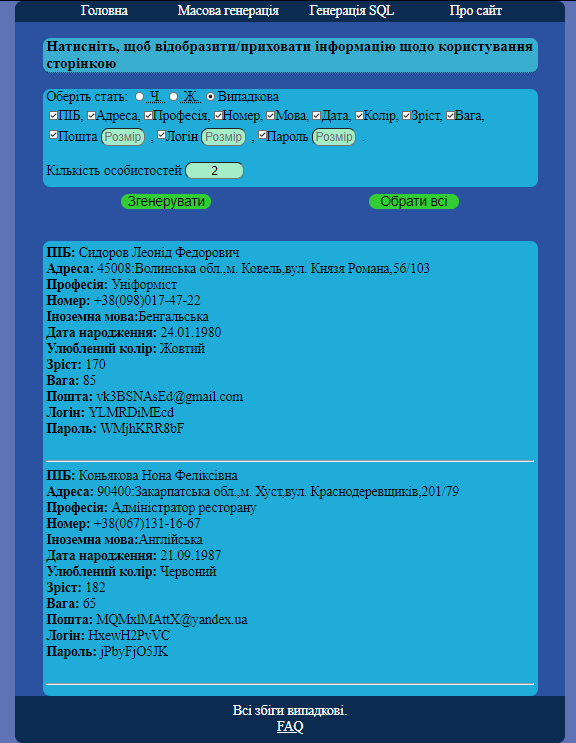
Додаток А.2



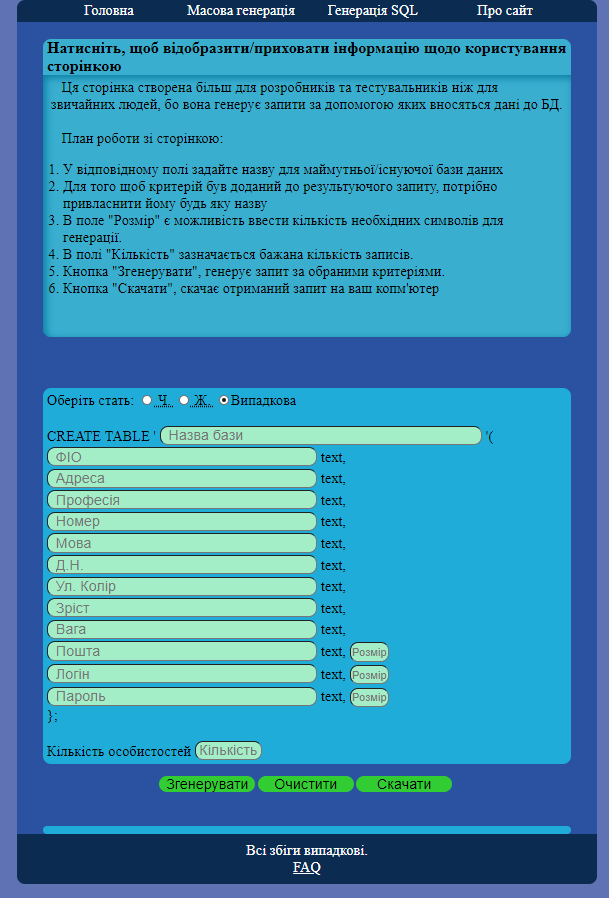
Додаток А.3



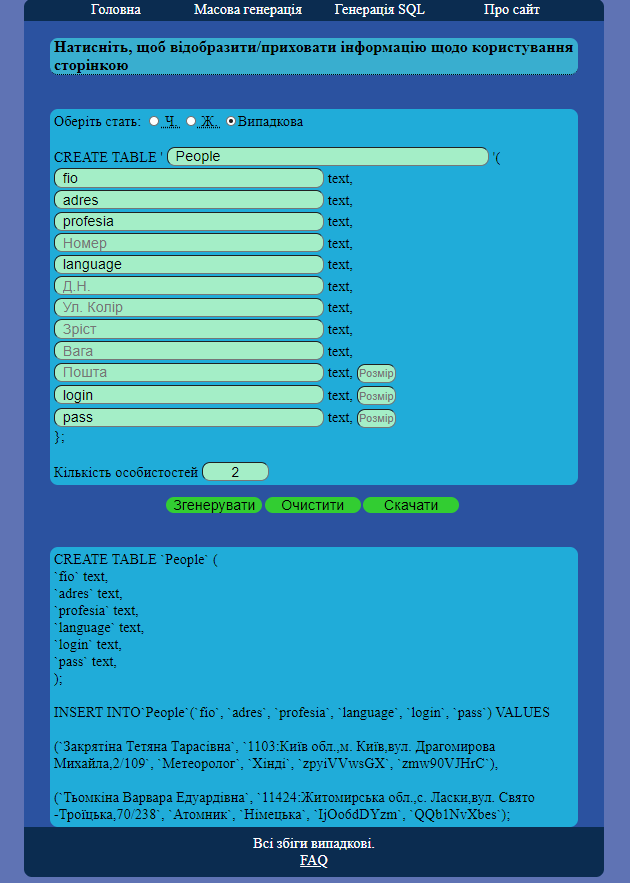
Додаток А.4



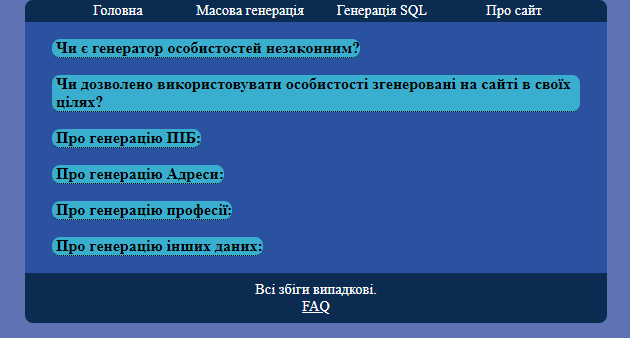
Додаток А.5



Додаток А.6



Додаток А.7



Додаток А.8



Додаток А

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | **13.02070849.51809 ПЗ** | | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  | Розробка інформаційної системи продуктового магазину  ER-модель | Літ. | | | Маса | | Масштаб |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  | |  |
| Розробив | | Костецький Д.В. |  |  |
| Перевірив | |  |  |  |
| Т.контр. | |  |  |  | Арк. 1 | | | | Аркушів 1 | |
|  | |  |  |  |  | НУ «Запорізька Політехніка»  КНТ-518 | | | | | |
| Н.контр. | |  |  |  |
| Затв. | |  |  |  |