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\_генератора особистостей \_\_

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

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 \_\_\_\_\_\_\_\_\_\_\_\_**«** Сучасних інтернет ехнологій»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Курс 3 Група КНТ-518 Семестр 5

**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):\_\_\_15.12.2020\_\_\_\_\_\_

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

3. Вихідні дані до проекту (роботи): Забезпечити введення та пошук даних \_\_

згідно з предметною областю; розробити додаток, за допомогою якого \_\_\_\_\_

забезпечити обробку інформації, створення форм, необхідних запитів \_\_\_\_\_\_

та звітів; доступ до інформації організувати за допомогою Web-\_\_\_\_\_\_\_\_\_\_\_\_

інтерфейсу.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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:\_23.09.2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + - 1. **КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсового  проекту (роботи) | Строк виконання етапів курсового проекту ( роботи ) | * + 1. Примітка |
| 1 | Отримання завдання на курсовий проект | **23.09.2020** |  |
| 2 | Опис та аналіз предметної області | **05.10.2020** |  |
| 3 | Аналіз технічного завдання | **10.10.2020** |  |
| 4 | Проектування структури та алгоритму роботи розроблюваної Web-системи | **20.10.2020** |  |
| 5 | Розробка компонентів Web-системи | **25.10.2020** |  |
| 6 | Розгортання та тестування розробленої Web-системи | **01.12.2020** |  |
| 7 | Складання пояснювальної записки | **25.11.2020** |  |
| 8 | Виконання графічної частини | **01.12.2020** |  |
| 9 | Представлення закінченої роботи на перевірку | **14.12.2020** |  |
| 10 | Захист курсової роботи | **15.12.2020** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

( підпис ) (прізвище та ініціали)

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

( підпис ) (прізвище та ініціали)

«18» Грудня 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 генератора особистостей

Лит.

Листів

НУ«Запорізька Політехніка»

Реценз.

Масса

Масштаб

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

WEB, СУБД, КЛІЄНТ, СЕРВЕР, СТРУКТУРА, HTML, CSS, JAVASCRIPT, PHP, MYSQL, ЗАПИТ, ТЕГ, СТИЛЬ,

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](#_Toc58792038)

[**1 Теоретична частина** 8](#_Toc58792039)

[**1.1 Аналіз предметної області** 8](#_Toc58792040)

[**1.2 Мова розмітки гіпертекстових сторінок HTML** 9](#_Toc58792041)

[**1.3 Основні технології** 12](#_Toc58792042)

[**2 РОЗРОБКА WEB-СИСТЕМИ** 19](#_Toc58792043)

[**2.1 Визначення вимог до системи.** 19](#_Toc58792044)

[**2.2 Складання технічного завдання на розробку системи.** 19](#_Toc58792045)

[**2.3 Придбання домену та хостингу.** 20](#_Toc58792046)

[**2.4** **Розробка ескізів дизайну.** 21](#_Toc58792047)

[**2.5 Верстка, програмування, тестування.** 23](#_Toc58792048)

[**2.6 Наповнення проекту** 26](#_Toc58792049)

[**2.7 Дослідна експлуатація та впровадження проекту.** 28](#_Toc58792050)

[**2.8 Оптимізація.** 28](#_Toc58792051)

[**2.9 Просування системи в мережі Інтернет.** 30](#_Toc58792052)

[**Висновок** 33](#_Toc58792053)

[**Перелік джерел посилань** 34](#_Toc58792054)

[Додаток А 35](#_Toc58792055)

[Додаток Б 39](#_Toc58792056)

[Додаток В 49](#_Toc58792057)

**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**

Головним етапом створення є повний аналіз предметної області для того шоб у майбутньому не зіштовхнутись з якоюсь проблемою, яку не передбачили перед тим я почали розробку сайту.

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.

Дизайн потрібно робити були простим зрозумілим та таким щоб відрізнявся від сайтів конкурентів, та легко запам’ятовувався користувачам, щоб вони потім повертались на сайт згодом.

Основне завданням з функціонального боку є збирання бази даних сайту з великою кількістю даних для генерації. Це пов’язано з тим 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 в навігації та користуванні, так як ним будуть користуватись як звичайні люди так і ті хто мають високий рівень навичок при роботі з комп’ютером.

## **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. Докладніше про ці технології:

**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.

Мінусами Angular.JS є те що в його основі лежить складна мова програмування і користувачі стикаються з помилками під час міграції між версіями.

**React**

React - це інструмент для створення користувацьких інтерфейсів. Його головне завдання - забезпечення виведення на екран того, що можна бачити на веб-сторінках. React значно полегшує створення інтерфейсів завдяки розподіленню кожної сторінки на невеликі фрагменти. Кожен виділений фрагмент сторінки, вважається компонентом.

Компонент React - це, якщо просто, то ділянка коду, який представляє частину веб-сторінки. Кожен компонент - це JavaScript-функція, яка повертає шматок коду, що представляє фрагмент сторінки. Для формування сторінки ми викликаємо ці функції в певному порядку, збираємо разом результати викликів і показуємо їх користувачеві. Всі компоненти, маленькі чи великі, можуть використовуватися повторно, навіть у різних проектах.

Мінусами React є невпорядкованість документації, великий вибір інструментів через які користувачі часто стають в ступор і для освоєння всіх його нюансів потрібно дуже багато часу.

Backend (працює на сервері обролює данні або відправляє результат до браузеру користувача) основна мова програмування яка використовуються - це PHP та фреймворки Django Node.js. Докладніше про ці технології:

**PHP**

Мова програмування РНР – це серверна мова за допомогою якого можна створювати Web-сайти, причому як невеликі, які складаються з однієї сторінки, так і гігантські системи, що використовують сотні і тисячі серверів. Він був створений для генерування HTML-сторінок на веб-сервері і роботи з базами даних. В даний час підтримується переважною більшістю хостинг-провайдерів. Входить в LAMP - «стандартний» набір для створення веб-сайтів (Linux, Apache, MySQL, PHP (Python або Perl)).

Плюси: маса розширень і бібліотек мови, застосування PHP допускається на абсолютно будь-яких серверах, потужність і гнучкість. PHP можна задіяти для роботи як з простими ресурсами, серед яких блоги і сайти-візитки, так і з корпоративними порталами, а також інтернет-магазинами. вільне програмне забезпечення. У купівлі ліцензії для використання PHP потреби немає. Безкоштовне застосування можливе як для комерційних, так і некомерційних проектів. Простота у вивченні. Досить одного тижня, щоб осягнути ази роботи з мовою.

Мінуси: передача об'єктів відбувається за значенням, створювати системні компоненти і десктопні програми за допомогою нього неможливо, розгортання додатків і налаштування сервера ускладнена глобальними параметрами налаштування. До всього іншого, вони впливають і на базис синтаксису PHP, невеликий набір інструментів для ведення діяльності з винятками. Потужними їх назвати важко, рівень безпеки додатків, створених на даній мові, невисокий.

**Node.js**

Node.js - це програмна платформа, яка робить JavaScript мовою загального призначення. Її також називають середовищем виконання JS. Вона вміє зв'язуватися з зовнішніми бібліотеками, викликати команди з коду і виконувати роль веб-сервера. Node.js також можна визначити як мова сценаріїв на стороні сервера. Якщо пояснювати простіше, цей інструмент додає до повністю фронтендової мови бекендовую частина, дозволяючи створювати з його допомогою не тільки веб-сайти, а й повноцінні програми, без задіяння браузера.

Документація Node.js розвинена недостатньо. До того ж Node.js не володіє основними бібліотеками та інструментами, а через велику кількість альтернатив не завжди зрозуміло, який варіант варто вибрати

Важкий обчислювальний запит блокує обробку інших завдань і уповільнює роботу програми, написаної на Node. Тому він не підходить для проектів, заснованих на науці про дані.

**Django**

Django - це веб-фреймворк написаний на мові Python. Він може використовуватися в основному з будь-яким фронтенд фреймворком, також є хорошим рішенням призначеним для швидкої, надійної і безпечної розробки динамічних сайтів та веб-додатків з постійно зростаючою аудиторією. З ростом популярності Python варто розглянути Django для серверної частини так як для нього написано багато бібліотек які спрощують життя. Веб-платформа Django ідеально підходить для створення масштабованих веб-додатків або веб-сайтів з постійно зростаючою аудиторією.

Django є великим і монолітним фреймворком. Це дозволяє спільноті розробляти сотні універсальних модулів і додатків, але крім того, кожна нова версія повинна бути назад сумісна з попередніми, щоб старі проекти продовжували працювати. Все це призводить до того, що фреймворк розвивається не так швидко.

На рахунок швидкості не можна сказати, що фреймворк сам по собі повільний. Однак, якщо неправильно спроектувати архітектуру, то вона в поєднанні з Python призведе до повільної роботи сайту або програми. Так само Django - не найкращий вибір для невеликих сайтів.

Проаналізувавши всі мови та фреймворки описані вище, я зрозумів що більше мені підходять наступні мови програмування: CSS, JS, PHP. Одним з поштовхів до цього вибору було те, що я маю достатні навички, та мав практику по роботі з ними по окремості.

CSS тому що ця мова краща для оформлення дизайну структурованих документів HTML, та має простій інтерфейс.

JS тому що цю мову подтримують всі браузери, та вона вважається най популярнішою мовою у світі.

PHP тому що для того щоб навчитись програмувати на цій мові не потрібно витрачати багато часу та більшість серверім підтримують цю мову.

Не менш важливим пунктом є обрати мову для розробки бази даних

Для реалізації бази даних буда обрана MySQL бо це найпоширеніша повноцінна серверна СКБД. Ця СКБД є безкоштовною, нові версії виходять постійно, розширюючи функціонал і покращуючи безпеку. У безкоштовні версії найбільший наголос робиться на швидкість і надійність. Вона має команди, за допомогою яких дані можна вилучати, сортувати, оновлювати, видаляти і додавати, є однією з найшвидших баз даних серед наявних . Кількість рядків у таблицях може досягати 50 млн.

Він працює на багатьох операційних системах, наприклад: Linux та Windows,  може працювати як на домашніх ПК, так і на потужних серверах. Вона підтримує великий набір різних інтерфейсів, зроблених користувачами. Ця система керування базами даних використовує стандартну мову SQL.

Мінусами цієї СКБД є те що за задумкою розробників в MySQL закладені деякі обмеження функціоналу, які іноді необхідні в особливо вимогливих додатках, проблеми з надійністю через деякі способи обробки даних тому вона іноді поступається іншим СУБД по надійності.

### **2 РОЗРОБКА WEB-СИСТЕМИ**

## **2.1 Визначення вимог до системи.**

Система повинна бути простою в користуванні. Не менш важливим є її швидкість завантаження, тобто сайт повинен працювати без великих затримок та використовувати не багато інтернет трафіку. Сервер повинен бути з потужним, щоб швидко оброблювати базу даних та відправляти результат до сайту, також важливе його географічне розташування та ціна. Код сайту повинен бути максимально оптимізований, щоб не сильно навантажував комп’ютер. Також не повинно бути не потрібних повторів у коді, так як кожен рядок що не використовується, буде просто збільшувати вагу документа, та через це буде зменшуватись швидкість завантаження. Ту частину де не використовується звертання до бази даних генерувати за допомогою javascript.

Найголовнішим для користувачів інтерфейс не повинен бути складним, а навпаки простим для сприйняття. На сайті повинно бути поле за допомогою якого людина ознайомитись з короткою інформацією про сторінку на якій знаходиться. Сайт повинен бути адаптований під мобільні пристрої.

## **2.2 Складання технічного завдання на розробку системи.**

Для людей с простими потребами повинна бути розроблена головна сторінка, яка повинена вміти генерувати або всі данні одразу або за натискання певної кнопки генерувати 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 буде створена таблиця, яка 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. Також для краси додати підсвічування кнопок при наведенні на них. Сайт повинен бути адаптованим під мобільні пристрої та протестований з різними розширеннями екрану.

## **2.3 Придбання домену та хостингу.**

Для опублікування в мережі інтернет будь-якого веб-сайту необхідно дві складові: домен і сервер, домен - унікальна адреса, за допомогою якої можна звертатися до сайту, і сервер (хостинг) - місце зберігання всіх файлів для коректної роботи сайту і обробки 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 днів 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 можемо завантажити до 5 сайтів та 5 гігабайтів інформації. За допомогою файлового менеджеру завантажується на хостинг код сайту, та базу даних в якій зберігаються дані. Якщо все було зроблено вірно, то при переході за доменом, який ми отримали при реєстрації повинен відкритись сторінка сайту. І за цим посиланням тепер буть-який користувач інтернету зможе перейти та переглянути завантажений сайт раніше.

## 2.4 **Розробка ескізів дизайну.**

Перед тим як розроблювати ескіз потрібно спочатку зрозуміти, що це таке та для чого його використовують при розробці сайтів і не тільки.

Ескіз - це найбільш точне відображення роботи майбутнього сайту в інтерактивній формі прототипу. З його допомогою можна оцінити не тільки зовнішній вигляд, але і функціонал різних елементів.

Перед тим як розпочати верстку сайту потрібно розробити ескізи всіх майбутніх сторінок, так як робити правки та додавати нові функції легше на стадії створення ескізів ніж по завершенню розробки сайту, бо в такій ситуації є велика ймовірність того що прийдеться переробити дуже велику частину сайту через нові правки.

Першим ескізом буде «Головна» сторінка, вона генерує дані як по одному полю, так і всі одразу за допомогою певних кнопок. Прототип цієї сторінки зображено у додатку А.1.

Другим ескізом буде сторінка «Масова генерація», вона генерує певну інформацію обраного поля за допомогою checkbox, або всієї інформації за допомогою кнопки "Обрати всі". Прототип цієї сторінки зображено у додатку А.2.

Другим ескізом буде сторінка «Генерація SQL», вона складається з полю в яке вноситься назва таблиці, та полів які слугують для того щоб якщо їм привласнено ім’я то вони будуть додані до результату генерації. Прототип цієї сторінки зображено у додатку А.3.

І останнім ескізом є сторінка «Про сайт» вона складається з тих питань, які можуть виникнути у користувача у ході користування сайтом. Прототип цієї сторінки зображено у додатку А.4.

## **2.5 Верстка, програмування, тестування.**

Верстка повинна відповідати розробленим шаблонам, на сайті повинна бути адаптивність під різні розширення екранів та підтримка кросбраузерності. Перед початком написання сайту було прийнято рішення розбити різні елементи сайту по текам, так як коли файли знаходяться в одній теці, то витрачається багато часу на те щоб знайти потрібний. На рисунку 2.1 зображено теки які використовуються моїм сайтом.



Рисунок 2.1 – Вміст головної теки сайту

В ході розробки сайту, були отримані наступні теки: css – для зберігання файлів з розширенням .css, які встановлюю стилі певним сторінкам, image – зберігає всі картинки які використовуються на сайті, в моєму випадку це іконка, яка відображається біля заголовка сайту, js – тека яка зберігає всі скрипти які написані мовою JavaScript, php – зберігає скрипти на писані мовою php які виконуються на стороні серверу, template – зберігає опис всіх сторінок сайту.

Для зручної розробки сайт було поділено на три частини: header, footer та body(основну частину сайту), зроблено це у зв’язку з тим що header та footer повторюється на кожній сторінці, тому вони були винесені до двох окремих файлі, і при заміні коду в одному з цих файлів, відображення зміниться на кожній сторінці сайту, це рішення дуже правильне, бо без нього потрібно було б вносити корективи до кожної сторінки сайту.

Реалізація header (menu.php) та footer приведена у додатках В.1 та В.2, їх стилі у додатках В.3 та В.4.

Для того щоб сайт мав адаптивність, у кожному файлі .css, було використано медіа запит «@media», який відстежує розмір екрану за допомогою властивостей max-width, тому коли він становиться менше 690 пікслелів сайту привласнюються нові інструкції CSS, а саме вмикається мобільна версія сайту.

На сайті також присутня анімація кнопок та елементів меню, яка виконується за допомогою CSS, якщо людина наводить курсор миші на будь-яку кнопку, то вона змінює колір та довкола неї з’являється світіння, якщо прибрати з неї курсор, то вона повернеться до початкового стану. Приклад коду приведено у лістингу 2.1.

Лістинг 2.1 – Анімація кнопки при наведенні на неї

.btn\_copy input[type=button] {

width: 55px;

outline: none;

border: none;

transition: background-color .3s, filter .3s;;

background-color: limegreen;

}

\* input[type=button]:hover {

filter: drop-shadow(0 0 6px #48CB8E);

background-color: rgb(102, 227, 166);

}

Розглянемо роботу сайту на прикладі «Головної» сторінки, код приведено у додатку В.5, налаштування стилів у додатку В.6, у додатку Б.1 зображено, як виглядає сторінка на комп’ютері. Розглянувши код, ми побачимо, що блоки header та footer імпортуються за допомогою виразу мови PHP «include», тобто код повністю береться з відповідного файлу та вставляється до коду головної сторінки. Під час натискання будь-якої кнопки, буде виконано інструкцію, яка привласнена її. Наприклад, натиснувши кнопку «Згенерувати ПІБ» буде створено get запит, який буде відправлено на сервер і оброблено за допомогою файлу gener.php (додаток В.7), результат буде повернуто на сторінку у вигляді JavaScript коду який в свою чергу встановить значення у відповідне поле. При натисканні на інші кнопки генерація відбувається за допомогою мови JavaScript, код приведено у лістингу В.8.

А тепер докладніше про те як генеруються ці значення. Після натискання на сайті однієї з кнопок “Згенерувати ПІБ”, «Згенерувати адресу», «Згенерувати професію» або «Згенерувати все» на сервер відправляється відповідний запит, файл починає його оброблювати. І в залежності від обраної кнопки виконується певна функція на мові php, всі функції мають схоже завдання, вони беруть з бази випадкові дані та у вигляді JavaScript коду повертають результат на сайт. Також вирішена проблема через яку зникали вже знегеровані данні на сайті, це було пов’язано з тим що коли сервер відправляє данні до сторінки, вона оновлюється, тому було прийняте рішення, щоб сайт відправляв на сервер значення вже згенерованих даних, та потім їх повертав з новим значенням. Під час натискання кнопок відмінних від описаних вище, генерація відбувається за допомогою мови JavaScript, для кожної кнопки створена відповідна кнопка, значення якої генерує сам браузер, тому ці поля генеруються швидше та не оновлюють сторінку. У лістингу 2.2 відображений код функції яка копіює значення з поля до буферу обміну користувача.

Лістинг 2.2 – Функція яка копіює значення поля в буфер обміну

function myFunction(name\_text) {

var copyText = document.getElementById(name\_text);

copyText.select();

document.execCommand("copy");

}

Інші сторінки генеруються майже аналогічно за відміною того що на сторінці «Масова генерація» відбір даних для генерації виконано у вигляді checkbox, та сервер генерує з отриманих даних значення, збирає всі разом та відправляє результат на сайт, сторінка «Генерація» працює аналогічно, лише замість checkbox сервер перевіряє пусте чи ні значення поля, якщо воно чимось заповнено, то це аналогічно тому, що checkbox активний, також на цій сторінці є нова кнопка, яка скачує результат у вигляді файлу на комп’ютер користувача. У лістингу 2.3 відображено код цієї функції

Лістинг 2.3 - Функція яка скачує результат у вигляді файлу

function download(obj) {

var str = JSON.stringify(obj);

var blob = new Blob([str], {

type: 'application/octet-stream'

});

var url = URL.createObjectURL(blob);

var link = document.createElement('a');

link.setAttribute('href', url);

link.setAttribute('download', 'from\_generator\_people.sql');

var event = document.createEvent('MouseEvents');

event.initMouseEvent('click', true, true, window, 1, 0, 0, 0, 0, false, false, false, false, 0, null);

link.dispatchEvent(event);

}

## **2.6 Наповнення проекту**

Проаналізувавши предметну область стало зрозуміло що потрібно створити базу даних, яка повинна складатися з таблиць name\_men, name\_women, surname\_man, surname\_woman, otchestvo\_man, otchestvo\_woman, profession та sity.

Поле ПІБ на сайті генерується за допомогою таблиць: name\_men, name\_women, surname\_man, surname\_woman, otchestvo\_man, otchestvo\_woman. Для того щоб на сайті генерувались особистості без повторень, потрібно додати багато різних даних до цих таблиць. У таблицях 2.1-2.6 показані атрибути відношень цих таблиць у базі даних. Поля професія та адреса на сайті генеруються за допомогою таблиць: profession та sity. У таблицях 2.7-2.8 показані атрибути відношень цих таблиць у базі даних. На рисунку 2.1 зображено графічне представлення бази даних у веб-додатку phpMyAdmin.

Таблиця 2.1 – Визначення атрибутів відношення name\_men

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_name | лічильник | Not null | Первинний ключ |
| name | Текст | Not null |  |

Таблиця 2.2 – Визначення атрибутів відношення name\_women

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_name | лічильник | Not null | Первинний ключ |
| name | Текст | Not null |  |

Таблиця 2.3 – Визначення атрибутів відношення surname\_men

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_surname | лічильник | Not null | Первинний ключ |
| surname | Текст | Not null |  |

Таблиця 2.4 – Визначення атрибутів відношення surname\_women

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_surname | лічильник | Not null | Первинний ключ |
| surname | Текст | Not null |  |

Таблиця 2.5 – Визначення атрибутів відношення otchestvo\_men

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_otchestvo | лічильник | Not null | Первинний ключ |
| otchestvo | Текст | Not null |  |

Таблиця 2.6 – Визначення атрибутів відношення otchestvo\_women

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_otchestvo | лічильник | Not null | Первинний ключ |
| otchestvo | Текст | Not null |  |

Таблиця 2.7 – Визначення атрибутів відношення profession

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| kod\_professii | лічильник | Not null | Первинний ключ |
| professia | Текст | Not null |  |

Таблиця 2.8 – Визначення атрибутів відношення sity

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип даних | Обов’язковість | Ключовий атрибут |
| num | лічильник | Not null | Первинний ключ |
| region | Текст | Not null |  |
| title | Текст | Not null |  |
| zip | Числовий | Not null |  |
| street | Текст | Not null |  |

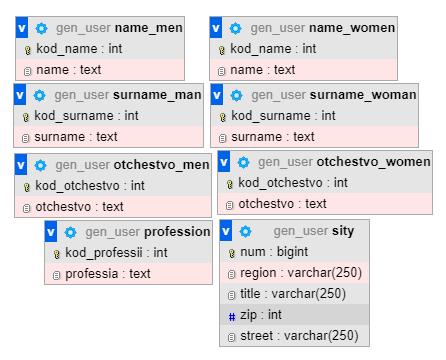


Рисунок 2.1 – Зображення таблиць розробленої бази даних

## **2.7 Дослідна експлуатація та впровадження проекту.**

Після запуску створеної Web-системи ми отримаємо домене ім’я за допомогою якого будь який користувач інтернету може перейти по ньому та осинитись на цьому сайті. Моєму сайту було привласнене домене ім’я <http://a0492513.xsph.ru/>. При перегляді змісту сайту з комп’ютера він буде виглядати як зображено у додатках Б.1-Б.8, а з мобільних пристроїв з різними розширеннями екрану та орієнтації зображено у додатку Б.9.

## **2.8 Оптимізація.**

Семантичне ядро - це набір слів і словосполучень, які користувачі вводять для пошуку необхідної їм інформації. Їх називають ключові слова і ключові фрази відповідно або, коротше, ключі.

Про аналізувавши багато ключових запитів були виділені основні, які зображені на рисунку 2.2, тому створення ключових слів було створено відштовхуючись від цих запитів.



Рисунок 2.2 – семантичне ядро сайту

На розробленому сайті кожна сторінка має свої ключові слова, заголовки та опис. Наприклад для головної сторінки були обрані такі пункти:

Ключові слова: генератор логінів,генератор особистостей, генератор паролів, генератор імен, генератор паролів онлайн,

згенерувати пароль, генерація паролів, генератор логіна, генератор пошт, випадкова пошта, рандомна адреса України, пошта, логін, Україна, адреса, ПІБ, українською.

Заголовок: Генератор особистості - головна

Опис: Сайт для швидкого створення випадкових особистостей, який буде корисний розробникам і тестувальникам для налагодження і тестування сервісів.

На рисунку 2.3 зображено як сайт представляється іншим, якщо його відправити повідомленням за допомогою месенджеру Telegram.

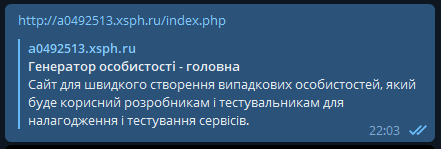


Рисунок 2.3 – Відображення інформації про сайт у месенджері Telegram

## **2.9 Просування системи в мережі Інтернет.**

Сайт був проіндексований та доданий до пошукової системи Google за допомогою сайту Google Search Console. Індексація сторінки - це процес збору, перевірки та внесення інформації про контент сайту до бази пошукових систем. Вона виконується в автоматичному режимі за допомогою роботів, але для того щоб вони про індексували сайт потрібно створити 2 файли: sitemap.xml та robots.txt.

Файл sitemap.xml- це файл з посиланнями на сторінки сайту, який повідомляє пошуковим системам про актуальну структуру (Карту) сайту. Sitemap в основному використовують, якщо на сайті: велика кількість сторінок, окремі сторінки без навігаційних посилань, глибока вкладеність, Також цей файл можна створити автоматично за допомогою сайту <https://majento.ru>.

Файл robots.txt надає пошуковим роботам інформацію про те, які сторінки і файли на вашому сайті можна обробляти, а які не можна. В першу чергу це інструмент, який дозволяє скоротити кількість запитів до сайту і тим самим знизити навантаження на нього.

Перевірити індексацію сайту можна декількома способами:

1. Виконати звичайний пошук за допомогою ключових слів. Результат можна побачити на рисунку 2.4.

2. Через оператор site: в Google. Результат можна побачити на рисунку 2.5.

3. За допомогою сайту Google Search Console у вкладці «Перевірка URL». Результат можна побачити на рисунку 2.6.

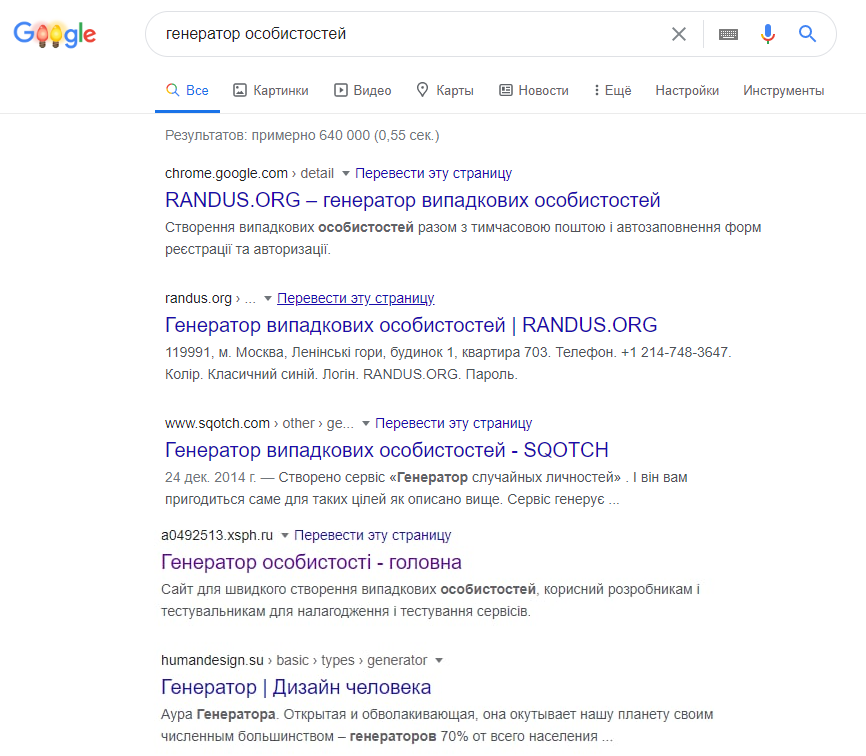
****

Рисунок 2.4 – відображення сайту у пошуковій системі Google

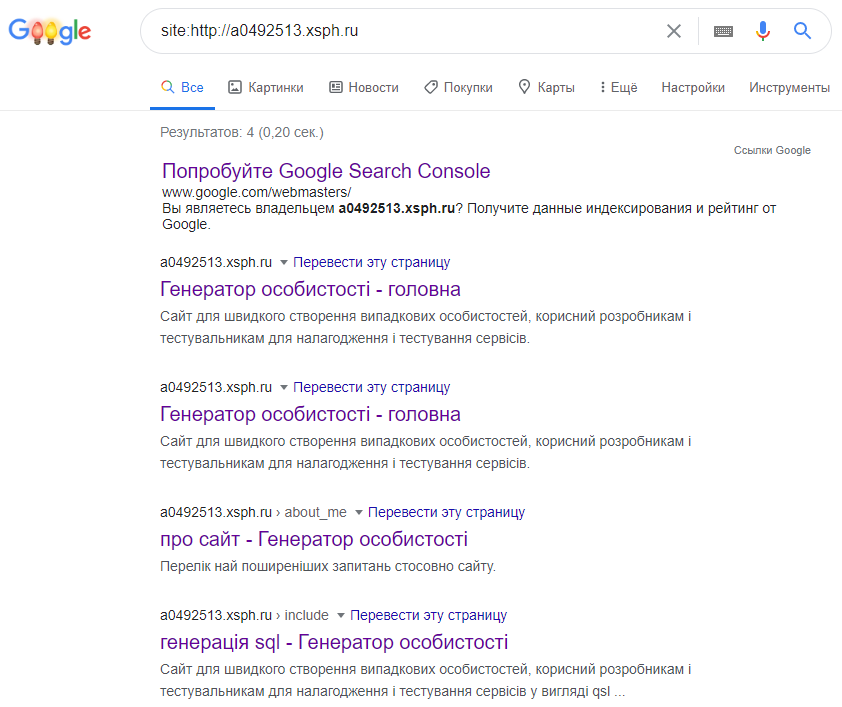


Рисунок 2.5 – Результати індексації за основним доменом за допомогою оператору site

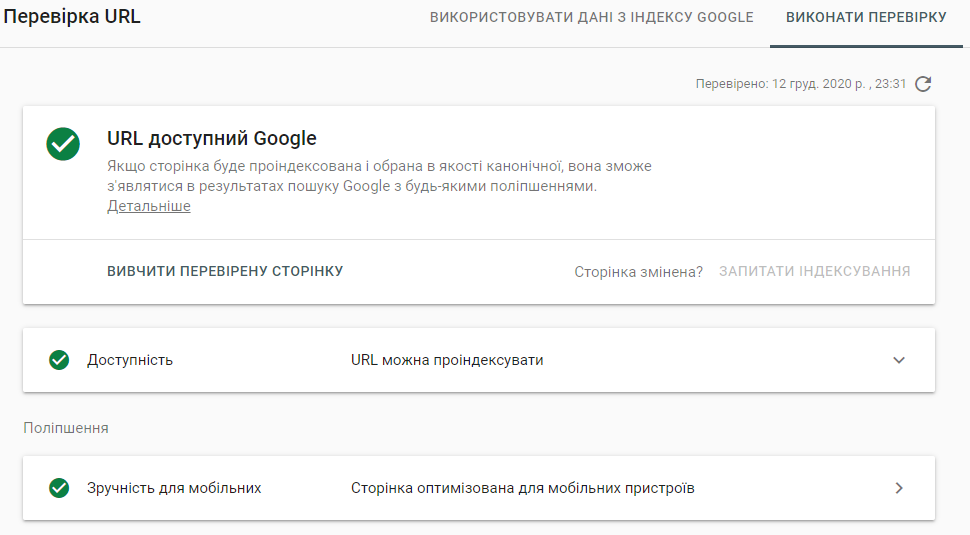


Рисунок 2.6 – перегляд інформації про індексацію сайту за допомогою

Google Search Console

### **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 були розроблені та переведені в ескіз структура і дизайн веб-сайту, який простий у використанні і відповідає всім вимогам.

В ході роботи був створений сайт для генерації особистості, який повністю відповідає поставленому технічному завданню, має інтуїтивно зрозумілий інтерфейс, оформлений в приємній колірній гамі, а саме най головніше має якісну адаптацію під різні розміри екрану. Генерація інформацій виконується швидко, без затримок, кожна сторінка має детальну інформація про користування нею. Ввесь сайт був перевірений за допомогою сторонніх сайтів валідаторів, та повністю пройшов перевірку. Також була ведена перевірка на SEO, яку сайт також пройшов на відміно.

Результатом курсової роботи було отримано сайт, який повністю оптимізований, має своє домене ім’я, налаштоване семантичне ядро, підключену індексацію. Тому він легко може конкурувати з сайтами конкурентів.

### **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**

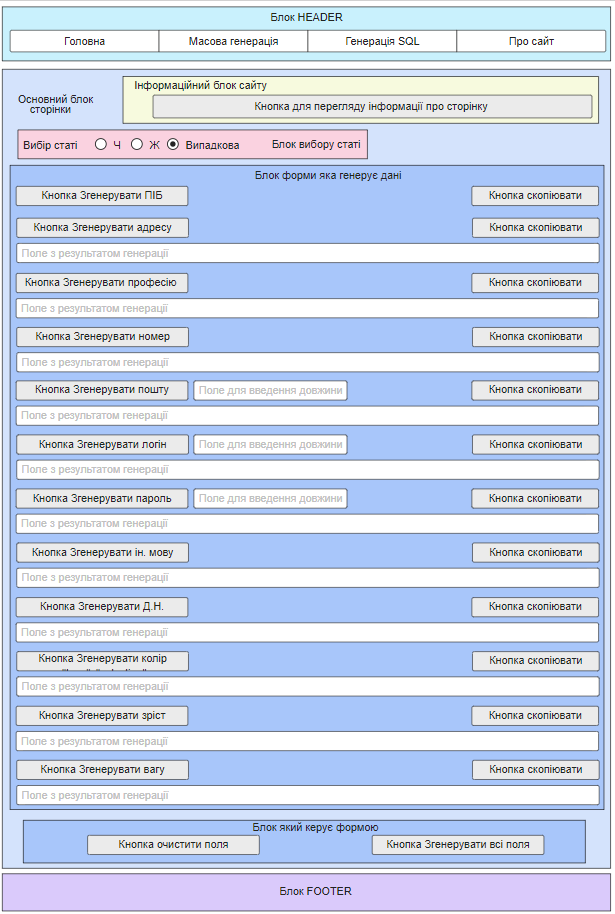


Рисунок А.1 – Ескіз головної сторінки

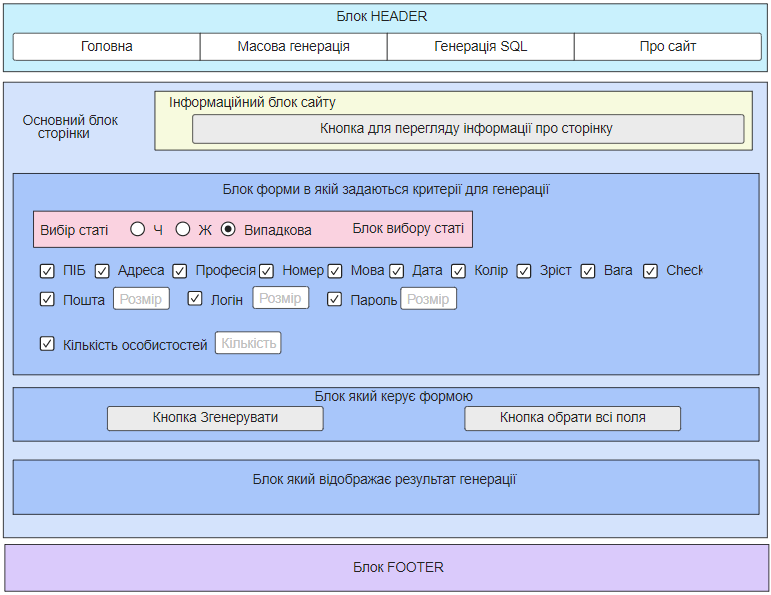


Рисунок А.2 - Ескіз сторінки «Масова генерація»

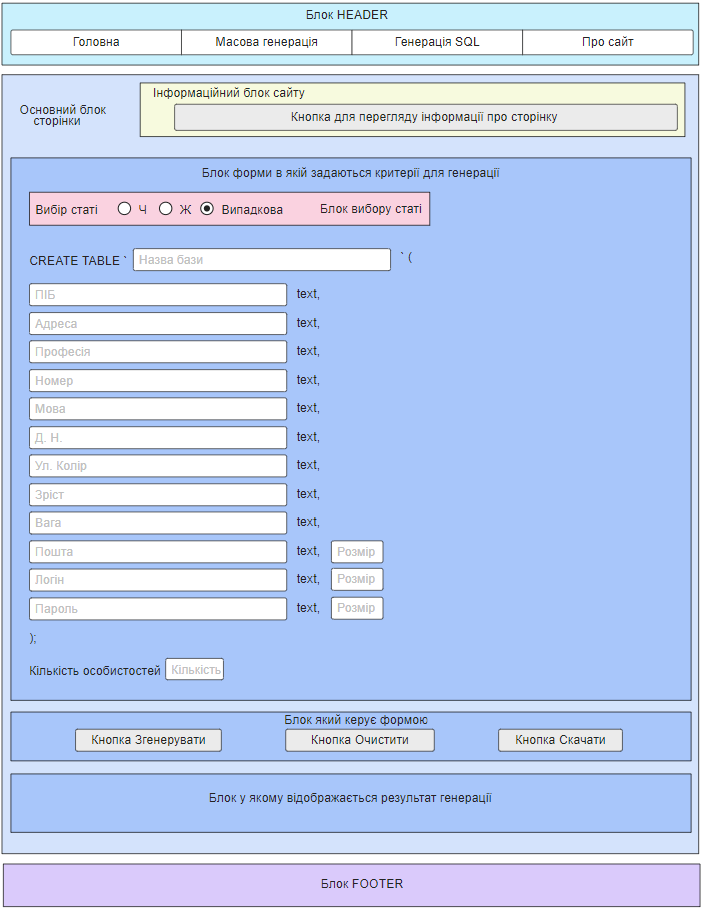
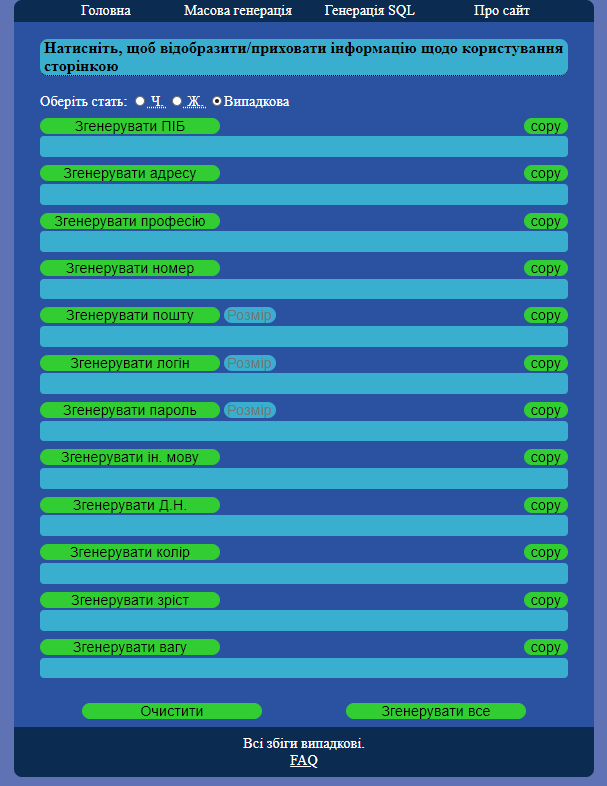
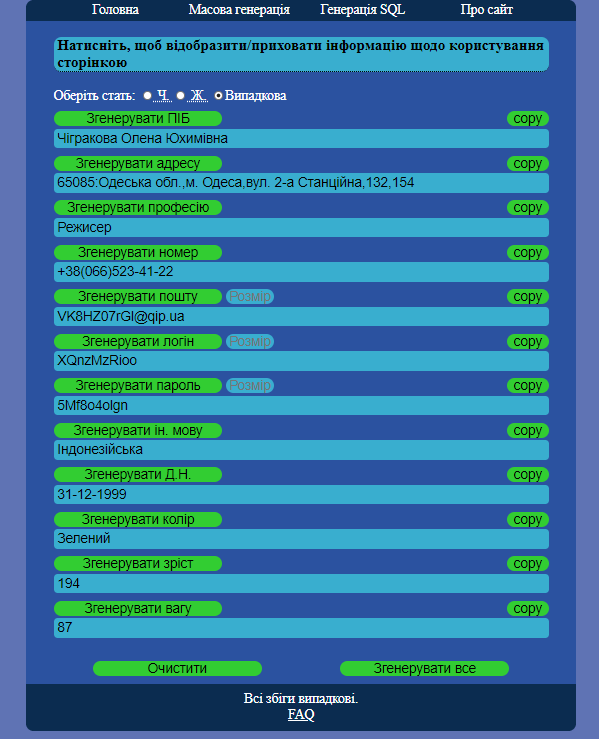


Рисунок А.3 Ескіз сторінки «Генерація SQL»

 Рисунок А.4 Ескіз сторінки «Про сайт»

### Додаток Б

 Рисунок Б.1 – Головна сторінка сайту

 Рисунок Б.2 – Головна сторінка сайту з згенерованими значеннями

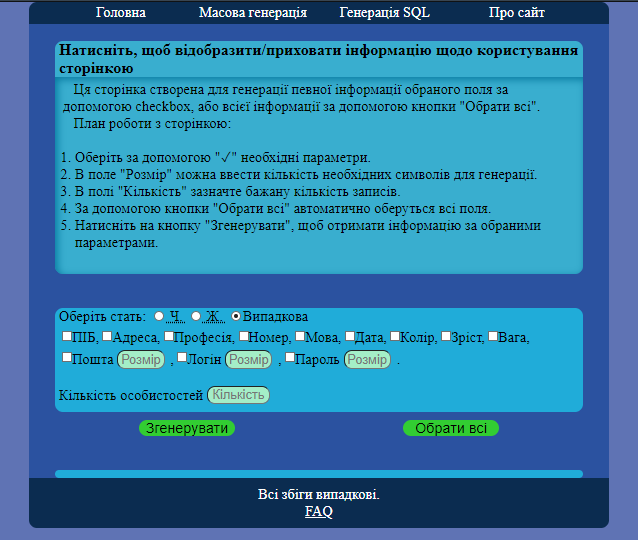


Рисунок Б.3– Сторінка «Масової генерації»

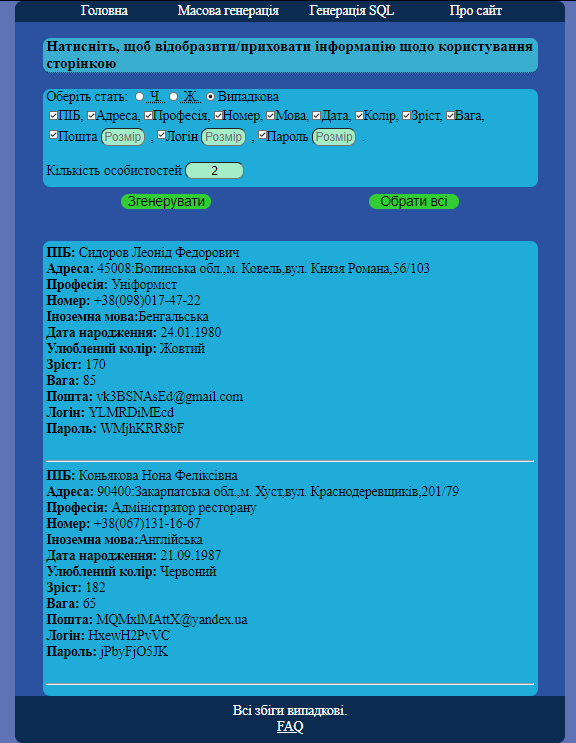


Рисунок Б.4– Сторінка «Масової генерації»

з згенерованими даними

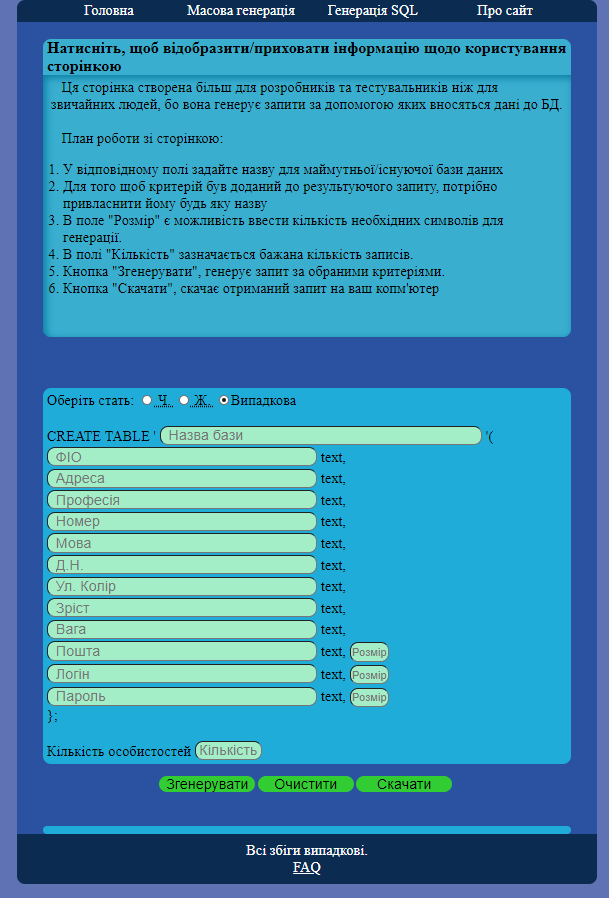


Рисунок Б.5 – Сторінка «Генерації SQL»

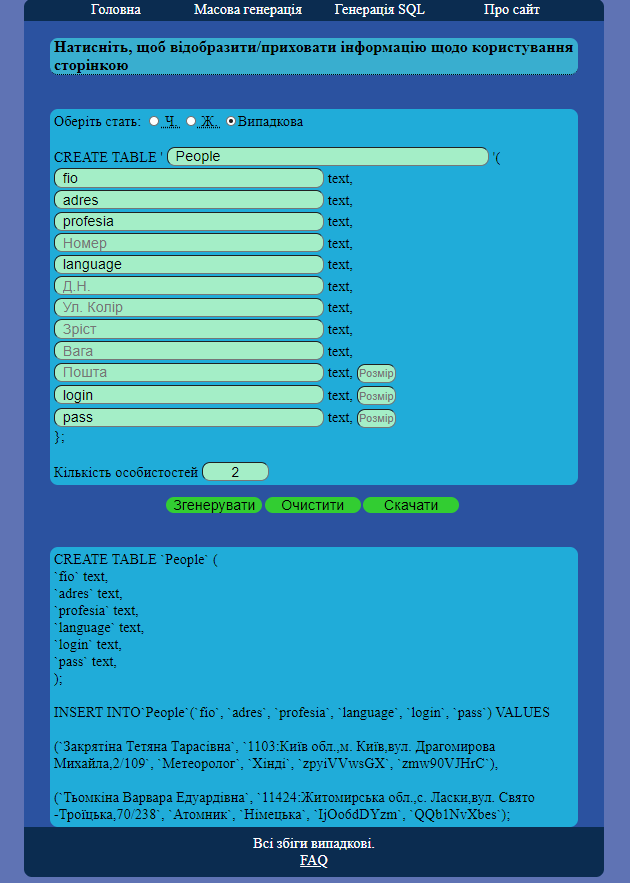


Рисунок Б.6– Сторінка «Генерації SQL»

З згенерованими даними

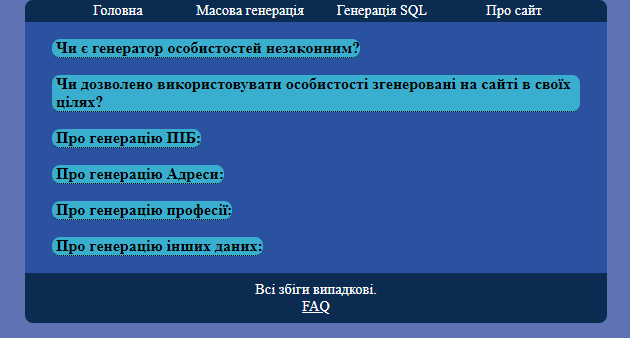


Рисунок Б.7– Сторінка «Про сайт»



Рисунок Б.8 – Сторінка «Про сайт»

з відкритими випадаючими текстами



Рисунок Б.9 – Відображення сайту

з різними розмірами екрану та орієнтацією

### Додаток В

Лістинг В.1

<ul id="head\_main">  
 <li><a href="/index.php">Головна</a></li>  
 <li><a href="/template/include/selective\_generation.php">Масова генерація</a></li>  
 <li><a href="/template/include/sql\_geneartor.php">Генерація SQL</a></li>  
 <li><a href="/template/include/about\_me.php">Про сайт</a></li>  
</ul>

Лістинг В.2

<div id="footer">

Всі збіги випадкові.<br>

<a href="/template/include/about\_me.php">FAQ</a>

</div>

Лістинг В.3

#head\_main {

font-size: 18px;

background: rgb(11, 44, 80);

border-radius: 10px 10px 0 0;

}

#head\_main {

display: flex;

padding-left: 2%;

padding-right: 2%;

width: 660px;

margin: auto;

}

#head\_main li {

display: inline-block;

width: 25%; /\* 4 пункта меню, то есть 100%/4 \*/

text-align: center;

}

#head\_main a {

display: block;

padding: 3px 0;

color: #fff;

text-decoration: none;

transition: background-color .3s, filter .3s;

}

#head\_main a:hover {

filter: drop-shadow(0 0 6px #6091AC);

border-radius: 10px 10px 0 0;

background: rgb(96, 145, 172);

}

@media (max-width: 690px) {

#head\_main {

display: flex;

width: 96%;

text-align: center;

align-content: center;

align-items: center;

}

#head\_main li {

height: 100%;

}

#head\_main a {

display: flex;

justify-content: center;

justify-items: center;

align-items: center;

height: 100%;

}

}

Лістинг В.4

#footer {

margin: 0px auto 5px auto;

text-align: center;

justify-content: center;

width: 660px;

background: rgb(11, 44, 80);

border-radius: 0 0 10px 10px;

padding: 10px 2% 10px 2%;

color: #fff;

}

#footer a{

color: white;

}

@media (max-width: 690px){

#footer {

width: 96%;

}}

Лістинг В.5

<!doctype html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<title>Генератор особистості - головна</title>

<meta name="description" content="Сайт для швидкого створення випадкових особистостей, корисний розробникам і тестувальникам для налагодження і тестування сервісів.">

<meta name="keywords" content="генератор логінів, генератор паролів, рандомайзер, генератор паролів онлайн,

згенерувати пароль, генерація паролів, генератор логіна, генератор пошт, випадкова пошта,

рандомний адресу України, пошта, логін, Україна, адреса, ПІБ, українською"/>

<meta name="language" content="uk">

<link rel="stylesheet" href="/css/menu.css">

<link rel="stylesheet" href="/css/footer.css">

<link rel="stylesheet" href="/css/index.css">

<link rel="stylesheet" href="/css/drop\_text.css">

<script src="js/generate.js"></script>

<link rel="shortcut icon" href="/image/ico.ico" type="image/x-icon">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<header>

<?php

include "template/interface/header.php";

?>

</header>

<FORM action="index.php" method=get>

<div class="generator">

<H1 class="h1\_hider">

Генератор особистості - головна

</H1>

<input name="all" type="hidden" value="find">

<br>

<div class="demo">

<input type="checkbox" id="hd-1" class="hide"/>

<label for="hd-1"> Натисніть, щоб відобразити/приховати інформацію щодо користування сторінкою</label>

<div>

&nbsp;&nbsp;&nbsp;Навіщо вигадувати дані, якщо сайт зробить це за вас?<br>

&nbsp;&nbsp;&nbsp;Сайт було розроблено для швидкого створення випадкових особистостей.<br>

&nbsp;&nbsp;&nbsp;Цей сайт буде корисний розробникам і тестувальникам для налагодження і тестування

сервісів.

<br>&nbsp;&nbsp;&nbsp;План роботи зі сторінкою:

<ol>

<li> Оберіть поле яке вас цікавить</li>

<li> Натисніть на кнопку "Згенерувати ..."</li>

<li> Після натискання, це поле буде згенероване</li>

<li> Для комфортної роботи є кнопка "copy" яка скопіює значення поля в буфер</li>

<li> Для генерації всіх полів натисніть внизу кнопку "Згенерувати все"</li>

<li> В поле "Розмір" можна ввести кількість необхідних символів для генерації</li>

</ol>

<br>

&nbsp;&nbsp;&nbsp;Для генерації декількох значень одного типу перейдіть на сторінку "Масова

генерація", яка знаходиться вгорі сайту.<br>

&nbsp;&nbsp;&nbsp;Якщо вам потрібно згенерувати запит для внесення даних до вашої БД, то перейдіть

на сторінку "Генерація SQL"

</div>

<br>

<br>

</div>

<div class="radio\_text">

<a>Оберіть стать:</a>

<input type="radio" name="course" value="1"/><abbr title="Чоловіча"> Ч. </abbr>

<input type="radio" name="course" value="2"/><abbr title="Жіноча"> Ж. </abbr>

<input type="radio" name="course" checked value="3"/>Випадкова

</div>

<div class="fio">

<input type="submit" name="fio\_btn" id="fio\_btn" value="Згенерувати ПІБ">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('fio\_text')"></div>

<input type="text" name="fio\_text" id="fio\_text"/>

</div>

<div class="address">

<input type="submit" name="address\_btn" id="address\_btn" value="Згенерувати адресу">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('address\_text')"></div>

<input type="text" name="address\_text" id="address\_text"/>

</div>

<div class="profession">

<input type="submit" name="profession\_btn" id="profession\_btn" value="Згенерувати професію">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('profession\_text')"></div>

<input type="text" name="profession\_text" id="profession\_text"/>

</div>

<div class="nomer">

<input type="button" value="Згенерувати номер" onclick="gen\_nomer()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('Nomer')"></div>

<input type="text" name="Nomer" id="Nomer"/>

</div>

<div class="email">

<div class="email\_num">

<input type="button" name="Email\_btn" id="Email\_btn" value="Згенерувати пошту"

onclick="gen\_email()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('Email\_txt')"></div>

<input type="text" name="Email\_num" id="email\_num" placeholder="Розмір" maxlength="2">

</div>

<input type="text" name="Email\_txt" id="Email\_txt"/>

</div>

<div class="login">

<div class="login\_num">

<input type="button" name="login\_btn" id="login\_btn" value="Згенерувати логін" onclick="gen\_login()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('login\_txt')"></div>

<input type="text" name="login\_num" id="login\_num" placeholder="Розмір" maxlength="2">

</div>

<input type="text" name="login\_txt" id="login\_txt"/>

</div>

<div class="password">

<div class="password\_num">

<input type="button" name="password\_\_btn" id="password\_btn" value="Згенерувати пароль"

onclick="gen\_password()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('password\_txt')"></div>

<input type="text" name="password\_num" id="password\_num" placeholder="Розмір" maxlength="2">

</div>

<input type="text" name="password\_txt" id="password\_txt"/>

</div>

<div class="lang">

<input type="button" name="lang\_btn" id="lang\_btn" value="Згенерувати ін. мову" onclick="gen\_lang()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('lang\_text')"></div>

<input type="text" name="lang\_text" id="lang\_text"/>

</div>

<div class="date">

<input type="button" name="date\_btn" id="date\_btn" value="Згенерувати Д.Н."

onclick="gen\_randomDate()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('date\_text')"></div>

<input type="text" name="date\_text" id="date\_text"/>

</div>

<div class="color">

<input type="button" name="color\_btn" id="color\_btn" value="Згенерувати колір" onclick="gen\_color()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('color\_text')"></div>

<input type="text" name="color\_text" id="color\_text"/>

</div>

<div class="height">

<input type="button" name="height\_btn" id="height\_btn" value="Згенерувати зріст" onclick="gen\_height()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('height\_text')"></div>

<input type="text" name="height\_text" id="height\_text"/>

</div>

<div class="weight">

<input type="button" name="weight\_btn" id="weight\_btn" value="Згенерувати вагу" onclick="gen\_weight()">

<div class="btn\_copy"><input type="button" value="copy" onclick="myFunction('weight\_text')"></div>

<input type="text" name="weight\_text" id="weight\_text"/>

</div>

<br>

<div class="nav\_btn">

<input type="reset" id="nav\_reset" name="Reset" value="Очистити">

<input type="submit" id="nav\_submit" name="all\_gener" value="Згенерувати все"

onClick="document.location = '?id=1'"/>

</div>

</div>

</FORM>

<footer>

<?php

include "template/interface/footer.php";

?>

</footer>

</body>

<?

include "php/gener.php";

?>

</html>

Лістинг В.6

body {

margin: 0px;

background-color: #5f73b4;

}

.h1\_hider {

display: none;

}

\* {

font-size: 18px;

padding: 0;

outline: none;

}

.radio\_text {

color: white;

}

\* input[type=text] {

margin-top: 2px;

outline: none;

padding: 3px 0 3px 0;

border: none;

width: calc(100%);

border: solid 1px #6a6a6a;

background-color: rgb(57, 174, 207);

text-indent: 5px;

border-radius: 5px 5px 5px 5px;

border: none;

}

\* input[type=button], \* input[type=submit], \* input[type=reset] {

background-color: limegreen;

margin-top: 10px;

width: 225px;

border-radius: 10px 10px 10px 10px;

outline: none;

border: none;

transition: background-color .3s, filter .3s;

}

.btn\_copy {

float: right;

}

.btn\_copy input[type=button] {

width: 55px;

outline: none;

border: none;

transition: background-color .3s, filter .3s;;

background-color: limegreen;

}

\* input[type=button]:hover {

filter: drop-shadow(0 0 6px #48CB8E);

background-color: rgb(102, 227, 166);

}

\* input[type=submit]:hover {

filter: drop-shadow(0 0 6px #48CB8E);

background-color: rgb(72, 203, 142);

}

\* input[type=reset]:hover {

filter: drop-shadow(0 0 6px #48CB8E);

background-color: rgb(72, 203, 142);

}

.btn\_copy input[type=button]:hover {

filter: drop-shadow(0 0 6px #48CB8E);

background-color: rgb(72, 203, 142);

}

.login\_num [type=text], .password\_num [type=text], .email\_num [type=text] {

width: 65px;

height: 20px;

margin-right: 3px;

padding: 0;

text-align: center;

text-indent: 0;

border-radius: 10px 10px 10px 10px;

}

.generator {

width: 660px;

margin-left: auto;

margin-right: auto;

background-color: #2b52a0;

padding: 0 2% 10px 2%;

}

#nav\_reset, #nav\_submit {

margin-left: auto;

margin-right: auto;

}

.nav\_btn {

margin: 0;

display: flex;

}

@media (max-width: 690px) {

input[type="radio"] {

height: 15px;

width: 15px;

}

\* input[type=button], \* input[type=submit], \* input[type=reset] {

width: 195px;

}

.login\_num [type=text], .password\_num [type=text], .email\_num [type=text] {

width: 60px;

height: auto;

}

.generator {

width: 96%;

}

.btn\_copy {

outline: none;

}

.btn\_copy input[type=button] {

width: 60px;

}

#nav\_reset, #nav\_submit {

width: 420px;

margin-bottom: 20px;

}

}

Лістинг В.7

<?

//Подключение к Базе Данных

//$str\_bd = new mysqli("localhost", "a0492513\_gen\_user", "zZ774488559966", "a0492513\_gen\_user");

$str\_bd = mysqli\_connect("127.0.0.1", "root", "root", "gen\_user");

//Здесь генерируется ФИО с базы данных

function gen\_fio($gender)

{

global $str\_bd;

//Выбор пола для получения количества записей в базе данных

if ($gender == "1") {

$strSQL = "SELECT COUNT(`kod\_name`) as counter FROM `name\_men` UNION SELECT COUNT(`kod\_otchestvo`) as counter FROM `otchestvo\_men`

UNION SELECT COUNT(`kod\_surname`) as counter FROM `surname\_man`";

} elseif ($gender == "2") {

$strSQL = "SELECT COUNT(`kod\_name`) as counter FROM `name\_women`

UNION SELECT COUNT(`kod\_otchestvo`) as counter FROM `otchestvo\_women`

UNION SELECT COUNT(`kod\_surname`) as counter FROM `surname\_woman`";

}

//Выполнение запроса

$r = mysqli\_query($str\_bd, $strSQL);

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

$arr\_res\_fio[$i] = $f['counter']; // В массив заносим значения с количеством записей в базах (name, otchestvo, surname)

//Такое решение было принято для того что бы запросы выполнялись быстрее, если без UNION, то запрос выполнялся бы 2 секунды, а сейчас 0,002секунды

}

//Переменные для генерации рандомного фио

$rand\_name = random\_int(1, $arr\_res\_fio[0] - 1);

$rand\_otchestvo = random\_int(1, $arr\_res\_fio[1] - 1);

$rand\_surname = random\_int(1, $arr\_res\_fio[2] - 1);

if ($gender == "1") {

$strSQL = "SELECT `surname` , `name`, `otchestvo`

FROM `name\_men`, `otchestvo\_men`, `surname\_man`

WHERE kod\_name=" . $rand\_name . " and kod\_otchestvo=" . $rand\_otchestvo . " and kod\_surname=" . $rand\_surname;

} elseif ($gender == "2") {

$strSQL = "SELECT `surname` , `name`, `otchestvo`

FROM `name\_women`, `otchestvo\_women`, `surname\_woman`

WHERE kod\_name=" . $rand\_name . " and kod\_otchestvo=" . $rand\_otchestvo . " and kod\_surname=" . $rand\_surname;

}

$r = mysqli\_query($str\_bd, $strSQL);

//Получаем значения благодаря рандомным переменным

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

echo "<script> document.getElementById('fio\_text').value='" . $f['surname'] . " " . $f['name'] . " " . $f['otchestvo'] . "' </script>";

}

}

//Здесь генерируется адрес с базы данных

function gen\_address()

{

global $str\_bd;

$strSQL = "SELECT max(`num`) as counter\_address FROM `sity`";

$r = mysqli\_query($str\_bd, $strSQL);

//получение количества записей в базе данных

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

$rand\_count\_addres = $f['counter\_address'];

}

//Переменная для генерации рандомного адреса

$rand\_addres = random\_int(0, $rand\_count\_addres - 1);

$strSQL = "SELECT `region` as obl, `title` as gor, `zip` as ind , `street` as yli FROM `sity` WHERE `num`=" . $rand\_addres;

$r = mysqli\_query($str\_bd, $strSQL);

//Получаем значение благодаря рандомной переменной

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

echo "<script> document.getElementById('address\_text').value='$f[ind]:$f[obl] обл.,$f[gor],$f[yli],'+getRandomInt(0, 167)+','+getRandomInt(0, 357) </script>";

}

}

//Здесь генерируется профессия с базы данных

function gen\_profession()

{

global $str\_bd;

$strSQL = "SELECT max(`kod\_professii`) as counter\_profession FROM `profession`";

$r = mysqli\_query($str\_bd, $strSQL);

//получение количества записей в базе данных

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

$rand\_count\_profession = $f['counter\_profession'];

}

//Переменная для генерации рандомной профессии

$rand\_profession = random\_int(1, $rand\_count\_profession - 1);

$strSQL = "SELECT `professia` as profession FROM `profession` WHERE `kod\_professii`=" . $rand\_profession;

$r = mysqli\_query($str\_bd, $strSQL);

//Получаем значение благодаря рандомной переменной

for ($i = 0; $i < mysqli\_num\_rows($r); $i++) {

$f = mysqli\_fetch\_array($r);

echo "<script> document.getElementById('profession\_text').value='$f[profession]'</script>";

}

}

$gender = $\_GET['course']; //get запрос с значениями пола, если рандом, то присваиваем 1 или 2 (1- мужской, 2- женский)

if ($gender == "3") {

$gender = random\_int(1, 2);

}

$sum\_arr\_text = array('fio\_text', 'address\_text', 'profession\_text', 'Nomer', 'login\_txt', 'login\_num', 'password\_txt', 'password\_num', 'Email\_txt', 'email\_num', 'lang\_text', 'date\_text', 'color\_text', 'height\_text', 'weight\_text');

//Эта вся конструкция, что бы значения в форме не удалялись

foreach ($sum\_arr\_text as &$temp\_iter\_arr\_) {

if ($\_GET[$temp\_iter\_arr\_] != "")

echo "<script>document.getElementById('$temp\_iter\_arr\_').value='" . $\_GET[$temp\_iter\_arr\_] . "';</script>";

}

//Здесь генерируются полностью все значения в поля

if ($\_GET["all\_gener"] == 'Згенерувати все') {

gen\_fio($gender);

gen\_address();

gen\_profession();

echo "<script>gen\_all();</script>";

}

// А это у меня определение что нужно вызвать из БД

if ($\_GET["fio\_btn"] == 'Згенерувати ПІБ') {

call\_user\_func('gen\_fio', $gender);

}

if ($\_GET["address\_btn"] == "Згенерувати адресу") {

call\_user\_func('gen\_address');

}

if ($\_GET["profession\_btn"] == "Згенерувати професію") {

call\_user\_func('gen\_profession');

}

//Закрываем поток обмена с базой данных

mysqli\_close($str\_bd);

?>

Лістинг В.8

function getRandomInt(min, max) {

return Math.floor(Math.random() \* (max - min)) + min;

}

//Эта функция вызывает все остальные функции которые в свою очередь генерируют определённые значения

function gen\_all() {

gen\_nomer();

gen\_email();

gen\_lang();

gen\_randomDate();

gen\_login();

gen\_password();

gen\_color();

gen\_height();

gen\_weight()

}

function myFunction(name\_text) {

var copyText = document.getElementById(name\_text);

copyText.select();

document.execCommand("copy");

}

function gen\_nomer() {

window.res\_nomer = "";

window.res\_nomer += "+38"

window.res\_nomer += "("

let arr\_nomer = ['067', '068', '096', '097', '098', '066', '095', '099', '063', '093', '092', '094'];

let item = arr\_nomer[Math.floor(Math.random() \* arr\_nomer.length)];

window.res\_nomer += item;

/\* +38-\*\*\*-\*\*\*-\*\*-\*\* \*/

window.res\_nomer += ")"

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += "-";

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += "-";

window.res\_nomer += getRandomInt(0, 9);

window.res\_nomer += getRandomInt(0, 9);

document.getElementById("Nomer").value = window.res\_nomer;

}

function gen\_email() {

let length

if(document.getElementById('email\_num').value==""){

length = 10

}

else{

length=document.getElementById('email\_num').value;

}

let alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890";

let domain\_email = [ '@yandex.ua', '@rambler.ua', '@qip.ua', '@bigmir.net', '@ukr.net', '@hotmail.com', '@yahoo.com', '@gmail.com', '@mail.ua'];

let res\_email = "";

let i=0;

while (i < length) {

res\_email += alphabet[Math.floor(Math.random() \* alphabet.length)];

i++;

}

res\_email+=domain\_email[Math.floor(Math.random() \* domain\_email.length)];

document.getElementById("Email\_txt").value = res\_email;

}

function gen\_lang(){

let arr\_lang = ['Англійська', 'Хінді', 'Іспанська', 'французька', 'Китайська', 'Арабська', 'Бенгальська', 'Російська', 'Індонезійська', 'Португальська', 'Німецька', 'Польська', 'Арабська', 'Японська'];

let res\_lang = arr\_lang[Math.floor(Math.random() \* arr\_lang.length)];

console.log(res\_lang);

document.getElementById("lang\_text").value = res\_lang;

}

function gen\_color() {

let arr\_color = ['Червоний','Помаранчевий','Жовтий','Зелений','Блакитний','Синій','Фіолетовий'];

document.getElementById("color\_text").value = arr\_color[Math.floor(Math.random() \* arr\_color.length)];

}

function gen\_randomDate(date1='01.01.1970', date2='01/01/2002'){

function randomDate(start, end) {

return new Date(start.getTime()

+ Math.random() \* (end.getTime() - start.getTime()));

}

var myDate = randomDate(new Date(1974, 0, 1), new Date(2002,0,1));

document.getElementById("date\_text").value = ( ('0' + myDate.getDate()).slice(-2)+ '-'

+ ('0' + (myDate.getMonth() + 1)).slice(-2)+ '-'+ myDate.getFullYear()

);

}

function gen\_login() {

let length

if(document.getElementById('login\_num').value==""){

length = 10

}

else{

length=document.getElementById('login\_num').value;

}

let alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890";

let res\_login = "";

let i=0;

while (i < length) {

res\_login += alphabet[Math.floor(Math.random() \* alphabet.length)];

i++;

}

document.getElementById("login\_txt").value = res\_login;

}

function gen\_password() {

let length

if(document.getElementById('password\_num').value==""){

length = 10

}

else{

length=document.getElementById('password\_num').value;

}

let alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890";

let res\_password = "";

let i=0;

while (i < length) {

res\_password += alphabet[Math.floor(Math.random() \* alphabet.length)];

i++;

}

document.getElementById("password\_txt").value = res\_password;

}

function gen\_height() {

document.getElementById("height\_text").value = getRandomInt(160, 195);;

}

function gen\_weight() {

document.getElementById("weight\_text").value = getRandomInt(55, 90);

}