

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ТЕЛЕКОМУНІКАЦІЙ

Пояснювальна записка
до магістерської кваліфікаційної роботи

на тему: **«РОЗРОБКА, МОДЕРНІЗАЦІЯ ТА ІНФОРМАЦІЙНА ПІДТРИМКА
СУПРОВОДЖЕННЯ І НАПОВНЕННЯ БАЗИ ДАНИХ “УКРАЇНСЬКИЙ
МАРТИРОЛОГ ХХ СТОЛІТТЯ”**

Виконав: студент 6 курсу, групи САДМ-61
_____ спеціальності 124 Системний аналіз
_____ (шифр і назва спеціальності)

_____ Грищенко М.Т.

_____ (прізвище та ініціали)

Керівник

_____ (прізвище та ініціали)

Рецензент

_____ (прізвище та ініціали)

Нормоконтроль

_____ (прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
 ТЕЛЕКОМУНІКАЦІЙ

Кафедра	Системного аналізу
Ступінь вищої освіти	Магістр
Спеціальність	124 Системний аналіз
	(шифр і назва)

ЗАТВЕРДЖУЮ
 Завідувач кафедри
 Системного аналізу

Гордієнко Т.Б.

_____ 20__ року

**ЗАВДАННЯ
 НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Грищенко Максим Тарасович

1. Тема роботи: «Розробка, модернізація та інформаційна підтримка супроводження і наповнення бази даних “Український мартиролог ХХ століття”», керівник роботи Гордієнко Тетяна Богданівна, к.т.н., професор, затверджені наказом вищого навчального закладу від _____ 2021 року № _____.
2. Строк подання студентом роботи _____ 2021 р.
3. Вихідні дані до роботи:
 1. Бази даних.
 2. Веб-розробка.
 3. Веб-форма.
 4. Мови програмування.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
 1. Аналіз існуючої бази даних.
 2. Створення нової бази даних та веб-форми.
 3. Демонстрація результатів.
5. Графічна частина роботи представлена на _____ слайдах презентації.
6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Пошук та збір теоретичної інформації	01.11.21 – 03.11.21	
2	Опрацювання теоретичної інформації	04.11.21 – 06.11.21	
3	Розробка гіпотез	07.11.21 – 08.11.21	
4	Перевірка гіпотез	09.11.21 – 11.11.21	
5	Підготовка до експериментів	12.11.21 – 13.11.21	
6	Проведення експериментів	14.11.21 – 16.11.21	
7	Результати експериментів	17.11.21 – 20.11.21	
8	Порівняння, висновки	21.11.21 – 21.11.21	

Студент

 Грищенко М.Т
 (підпис) (прізвище та ініціали)

Керівник роботи

 Гордієнко Т.Б
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської кваліфікаційної роботи: стор. 77, рис. 65, табл. 9, дод. 10.

Об'єкт дослідження – база даних для зберігання оцифрованих документів.

Предмет дослідження – модернізація бази даних.

Мета роботи – отримання більш стабільної та оптимізованої бази даних отримання.

Методи дослідження – модернізація бази даних та розробка веб-форми для її наповнення.

У роботі проведено аналіз предметної області побудови баз даних. Отримано оцінки часу наповнення бази даних стандартними методами та за допомогою веб-форми.

Встановлено практичність застосування веб-форми для наповнення бази даних.

БАЗА ДАНИХ, ВЕБ-ФОРМА, ПРОГРАМУВАННЯ, ПОЛЕ ВВОДУ,
ВИПАДАЮЧИЙ СПИСОК

ЗМІСТ

ВСТУП	9
1. ТЕОРЕТИЧНІ ВІДОМОСТІ ...	10
1.1 База даних.....	10
1.1.1 Визначення поняття	10
1.1.2 Типи баз даних.....	10
1.1.3 Системи керування базами даних	11
1.1.4 Архітектури баз даних	12
1.2 Мови програмування для веб-форм	12
1.2.1 Основні мови програмування.....	12
1.2.2 PHP мова програмування	15
1.2.3 HTML мова програмування	15
1.2.4 JavaScript мова програмування.....	16
1.2.5 CSS мова програмування.....	16
2. АНАЛІЗ ТА СТВОРЕННЯ МОДЕРНІЗОВАНОЇ БАЗИ ДАНИХ.....	17
2.1 Аналіз існуючої бази даних.....	17
2.2 Розробка структури нової бази даних.....	20
2.3 Створення модернізованої бази даних.....	22
2.4 Аналіз веб-форми.....	35
2.5 Розробка інтерфейсу бази даних.....	36
2.6 Розробка функціональної частини веб-форми.....	45
3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ.....	55
3.1 Результати модернізації бази даних	55
3.2 Результати роботи створеної веб-форми	59
3.3 Рекомендації до отриманих результатів	65

ВИСНОВКИ	66
ПЕРЕЛІК ПОСИЛАНЬ	67
ДЕМОНТРАЦІЙНІ МАТЕРІАЛИ	69

ВСТУП

У сучасному світі бази даних відіграють дуже велику роль у всіх галузях життєдіяльності сучасної людини. Хоч це і не помітно для звичайної людини, але всі дані котрі використовуються нею для життя, наприклад банківські рахунки, картки в медичних закладах, паспортні данні чи новини в соціальних мережах, все це зберігається в базах даних.

База даних “Український мартиролог ХХ століття” являється підсумком роботи державних архівів, започаткованої ще 1989 р., яка прийняла нові обороти після прийняття Закону України «Про реабілітацію жертв політичних репресій в Україні» в 1991 році, коли почали складатися регіональні та тематичні бази даних на репресованих.

Інформація, що представлена базою даних “Український мартиролог ХХ століття” містить стислу інформацію про жертв політичних репресій 1920-1950-х: фотопортрет або декількох фото, ім’я жертви і короткого опису роду зайнять, дат життя, інформацію відносно реабілітації, посилаються на архів, архівно-слідчі справи та інші матеріали тримаються. Це в першу чергу центральні державні архіви і державні архіви областей, а також галузеві державні архіви СБУ та МВС України.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 База даних

1.1.1 Визначення поняття

База даних – сукупність спеціально організованих даних, що відображають предметну область призначеної бази. Будь-який впорядкований набір даних вважається базою даних.

За допомогою баз даних людина здобуває потужний інструмент для інформаційного обміну та зберігання даних у сучасному суспільстві.

1.1.2 Типи баз даних

Реляційні бази даних. Реляційні бази даних стали переважати в 1980-х роках. Дані в реляційній базі організовані у вигляді таблиць, що складаються із стовпців і рядків. Реляційна СУБД забезпечує швидкий і ефективний доступ до структурованої інформації.

Об'єктно-орієнтовані бази даних. Інформація в об'єктно-орієнтованій базі даних представлена у формі об'єкту, як в об'єктно-орієнтованому програмуванні.

Розподілені бази даних. Розподілена база даних складається з двох або більше за частини, розташовані на різних серверах. Така база даних може зберігатися на декількох комп'ютерах.

Сховища даних. Будучи централізованим сховищем для даних, сховище даних являє собою тип бази даних, спеціально призначеної для швидкого виконання запитів і аналізу.

Бази даних NoSQL. База даних NoSQL, або нереляційна база даних, дає можливість зберігати і обробляти неструктуровані або слабоструктуровані дані

(на відміну від реляційної бази даних, що задає структуру містяться в ній даних). Популярність баз даних NoSQL зростає в міру поширення і ускладнення веб-додатків.

Графові бази даних. Графова база даних зберігає дані в контексті сутностей і зв'язків між сутностями.

Бази даних OLTP. База даних OLTP-це база даних призначена для виконання бізнес-транзакцій, виконуваних безліччю користувачів.

1.1.3 Системи керування базами даних

Перші бази даних з'явилися дуже давно, як тільки з'явилася потреба в обробці великих масивів інформації та вибірки груп записів за певними ознаками. Для цього була створена структурована мова запитів SQL (Structured Query Language). Він заснований на потужній математичній теорії і дозволяє виконувати ефективну обробку баз даних, маніпулюючи не окремими записами, а групами записів.

Для управління великими базами даних та їх ефективної обробки розроблені СУБД (системи управління Базами даних). Практично в кожній СУБД крім підтримки мови SQL є також свій унікальний мову, орієнтований на особливості цієї СУБД і не переноситься на інші системи.

MySQL-це реляційна система управління базами даних з відкритим вихідним кодом на основі мови SQL. Вона була розроблена і оптимізована для веб-додатків і може працювати на багатьох платформах. Вона володіє всіма можливостями, які потрібні веб-розробникам. База даних MySQL призначена для обробки мільйонів запитів і тисяч транзакцій, тому її часто вибирають компанії електронної комерції, яким потрібно управляти великою кількістю грошових переказів. Гнучкість у міру необхідності-основна характеристика MySQL.

1.1.4 Архітектури баз даних

Базу даних можна представити як людинозорієнтовану та комп'ютерозорієнтовану-на модель даних, кожна з яких визначає своє середовище зберігання даних. Людинозорієнтована модель бази даних відображається в паперових документах і в пам'яті людини, а компютерозорієтована - на фізичних носіях ЕОМ.

Кожен рівень даних характеризується своїм описом (сприйняттям) елементів даних - зовнішній (інфологічну), концептуальний (дatalogічний) і внутрішній (фізичний) рівень. Ці рівні формують трирівневу архітектуру баз даних.

- перший тип-зв'язок "один-до-одного" (1: 1): в кожен момент часу кожному представнику сутності А відповідає 1 або 0 представників сутності в.
- другий тип-зв'язок "один-до-багатьох" (1: М): одному представнику сутності А відповідає 0,1 або кілька представників сутності в.
- між двома сутностями можливі зв'язки в обох напрямках, тому існують ще два типи зв'язків "багато-до-одного" (г.: 1) і "багато-до-мно-ти" (м: n).

1.2 Мови програмування для веб-форм

1.2.1 Основи мов програмування

Існують різні способи створення веб- форм. Найпростішим способом є використання вже готового продукту, який спеціалізується на тому, що пошаблонно дає змогу користувачеві створити форму. На даний момент існує велика кількість веб-сервісів, котрі дають можливість створити такі форми.

Головним недоліком таких форм є мала кількість шаблонів і їх погана варіативність. Іноді в задачі стоять незвичайні потреби, котрі не передбачені в

подібних сервісах, а навіть якщо потреби не високі, навіть найменше відхилення від початкової задачі, може привести до поганих наслідків у кінці, що погрузе невиконанням завдання та витіканню негативних наслідків із цього.

Тому для створення специфічних веб-форм найкращим варіантом буде, створення такої бази з нуля. Такий шлях значно складніший, так як потребує знань в деяких областях, а саме програмуванні. Потрібно мати розуміння як створюється веб-сторінка та як вона функціонує. Треба знати за допомогою яких інструментів можна надати формі такого виду, якого потребує задача. І немало важливо розробити АПИ, яке буде виконувати всі задачі, що поставленні перед веб-формою.

Існуючі мови програмування класифікують за чотирма основними групами: процедурні, об'єктно орієнтовані, функціональні та логічні. Дамо короткі визначення кожного підходу.

Процедурне програмування - таке програмування, коли програма відокремлена від даних і складається з послідовності команд, що обробляють дані. Дані як правило зберігаються у вигляді змінних. Весь процес обчислення зводиться до зміни їх вмісту.

Декларативні мови програмування-це мови оголошень і побудови структур. До них відносяться функціональні та логічні мови програмування. У цих мовах не проводиться алгоритмічних дій явно, тобто алгоритм не задається програмістом, а будується самою програмою. У декларативних мовах задається, проводиться побудова будь-якої структури або системи, тобто декларуються (оголошуються) якісь властивості створюваного об'єкта.

Об'єктно-орієнтоване програмування - в цих мовах змінні і функції групуються в так звані класи (шаблони). Завдяки цьому досягається більш високий рівень структуризації програми. Об'єкти, породжені від КЛАСІВ викликають методи (функції або процедури) один одного і змінюють таким чином стан властивостей (змінних)

Мережеві мови - мови, призначені для організації взаємодії віддалених комп'ютерів в інтенсивному інтерактивному режимі, а тому вони побудовані на

принципах інтерпретації, тобто порядкової, інтерактивної обробки рядків програмного коду, що описує деякий сценарій (скрипт) мережевої взаємодії комп'ютерів, тому часто вони називаються скриптовими мовами, хоча скриптові мови не обов'язково є мережевими, приміром, пакетні командні мови різних операційних середовищ.

Високорівнева мова програмування - мова програмування, розроблена для швидкості і зручності використання програмістом. Основна риса високорівневих мов-це абстракція, тобто введення смислових конструкцій, коротко описують такі структури даних і операції над ними, описи яких на машинному кодї (або іншому низькорівневому мовою програмування) дуже довгі і складні для розуміння.

Об'єктно-орієнтоване програмування представляє наступні переваги:

- зручний і простий спосіб введення нових понять, найбільш близьких до понять реального світу;
- проста розробка компонент багаторазового застосування;
- легка модифікованість (адаптованість) програм;
- локалізація властивостей і поведінки об'єктів в одному місці.

Кожен об'єкт характеризується станом (властивості об'єкта) і можливістю виконувати деякі дії (методи об'єкта). Наприклад, для об'єкта "накладна" це можуть бути змінні, в тому числі структури даних, в яких зберігаються значення реквізитів документа, зовнішній вигляд документа, значення рядків і т. д. Для виконання необхідних дій звертаються (викликають) до методів об'єкта, наприклад, друк, висновок на екран, додавання рядка і т. п.

Клас об'єктів - це узагальнене (абстрактне) опис безлічі однотипних об'єктів. Об'єкти є конкретними представниками свого класу, їх прийнято називати екземплярами класу. Наприклад, клас кішки-поняття абстрактне, а екземпляр цього класу мій кіт Філя - поняття конкретне. Основними поняттями ООП можна вважати-абстрагування, успадкування і поліморфізм.

Розділ програмування, в рамках якого створюються ПРОГРАМИ ІНТЕРНЕТ-додатків, що реалізують різноманітні Інтернет-технології називаються інтернет - програмуванням. Програми інтернет-додатків називаються скриптами.

Існує дві основні категорії скриптів-клієнтські, тобто виконувані на комп'ютері користувача і серверні, призначені для використання на інтернет-сервері.

Як правило, клієнтські скрипти виконуються за допомогою браузерів, спеціальних програм, за допомогою яких проглядаються інтернет-сторінки. Ці скрипти являють собою якісь команди на спеціалізованій мові програмування, які полягають серед загального коду web-сайту. Найпоширенішими клієнтськими мовами програмування є JavaScript, VBScript, ActionScript і Java.

1.2.2 PHP мова програмування

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробок і його код може впроваджуватися безпосередньо в HTML.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері і генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний вищенаведеному, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код його справив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

1.2.3 HTML мова програмування

HTML (Hypertext Markup Language) - мова гіпертекстової розмітки. Розмітки утворюються за допомогою тегів («міток») — це набір символів, розташованих в дужках. Сторінка складається з тегів, що виконують відповідні завдання. На самій сторінці ми не побачимо ці символи - вони перетворюються.

Це свого роду цеглини, які утворюють в результаті цілий будинок. Від того, як правильно розклали «цеглинки» чи ні, залежить форма сторінки.

1.2.4 JavaScript мова програмування

JavaScript-це мова програмування, яку використовують для написання frontend - і backend-частин сайтів, а також мобільних додатків.

Під поняттям Frontend мається на увазі розробка видимого для користувача інтерфейсу і всіх функцій, з якими він може взаємодіяти. По суті, коли ви переходите на будь-який сайт, то бачите там кнопки, текст, різну анімацію та інші складові-все це реалізовано за допомогою фронтенду.

Backend-це внутрішня частина продукту, яка знаходиться на сервері і прихована від користувачів. Backend-розробник робить так, щоб сервер сформував відповідь і відправив його на потрібний комп'ютер. Про дії в backend багато хто навіть не замислюється-для них сторінка завантажується сама по собі.

1.2.5 CSS мова програмування

CSS - це формальна мова, службовець для опису оформлення зовнішнього вигляду документа, створеного з використанням мови розмітки. Каскадні таблиці стилів працюють з HTML, але це зовсім інша мова. HTML структурує документ і впорядковує інформацію, а CSS взаємодіє з браузером, щоб надати документу оформлення.

2 АНАЛІЗ ТА СТВОРЕННЯ МОДЕРНІЗОВАНОЇ БАЗИ ДАНИХ

2.1 Аналіз існуючої бази даних

У теоретичній частині роботи було розглянуто поняття бази даних, способів керування нею та способи наповнення. Також доволі широко розкриті поняття та застосування мов програмування, що використовуються в базах даних.

На даний момент самостійної бази даних для внесення карток з відомостями про репресованих осіб немає, існують лише деякі таблиці з доданими даними, які знаходяться в чисті таблиць бази даних АФУ(Архівні фонди України).

Первинною задачею стоїть створення самостійної бази даних в яку буде перенесено всі дані зі старих таблиць та відформатовано для коректного відображення.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
lancestry	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
addresses	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
addresses_archives	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
afu_files	★ Обзор Структура Поиск Вставить Очистить Удалить	5 429	InnoDB	latin1_swedish_ci	368.0 КзБ	-
afu_records	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	latin1_swedish_ci	16.0 КзБ	-
alternative_date_records	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
archives	★ Обзор Структура Поиск Вставить Очистить Удалить	1 817	InnoDB	utf8mb3_unicode_ci	1.5 МБ	-
archives_en	★ Обзор Структура Поиск Вставить Очистить Удалить	1 817	InnoDB	utf8mb3_unicode_ci	1.5 МБ	-
arch_zakon	★ Обзор Структура Поиск Вставить Очистить Удалить	195	InnoDB	latin1_swedish_ci	144.0 КзБ	-
arch_zak_vid_doc	★ Обзор Структура Поиск Вставить Очистить Удалить	17	InnoDB	latin1_swedish_ci	32.0 КзБ	-
cardboardboxes	★ Обзор Структура Поиск Вставить Очистить Удалить	299	InnoDB	latin1_swedish_ci	48.0 КзБ	-
contacts	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
contacts_plus_addresses	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
controller	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_general_ci	16.0 КзБ	-
countries	★ Обзор Структура Поиск Вставить Очистить Удалить	141	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
definition_untruly	★ Обзор Структура Поиск Вставить Очистить Удалить	13	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
folders	★ Обзор Структура Поиск Вставить Очистить Удалить	5 282	InnoDB	utf8mb3_unicode_ci	2.5 МБ	-
fonds	★ Обзор Структура Поиск Вставить Очистить Удалить	~397 065	InnoDB	utf8mb3_unicode_ci	39.6 МБ	-
fonds_names	★ Обзор Структура Поиск Вставить Очистить Удалить	~538 363	InnoDB	utf8mb3_unicode_ci	141.7 МБ	-
fonds_parts	★ Обзор Структура Поиск Вставить Очистить Удалить	~304 646	InnoDB	utf8mb3_unicode_ci	110.6 МБ	-
fonds_types	★ Обзор Структура Поиск Вставить Очистить Удалить	31	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
founder	★ Обзор Структура Поиск Вставить Очистить Удалить	1 747	InnoDB	utf8mb3_unicode_ci	208.0 КзБ	-
information_fixing_material	★ Обзор Структура Поиск Вставить Очистить Удалить	10	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
inventories	★ Обзор Структура Поиск Вставить Очистить Удалить	113	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
koatuu	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb3_general_ci	16.0 КзБ	-
language	★ Обзор Структура Поиск Вставить Очистить Удалить	544	InnoDB	utf8mb3_unicode_ci	80.0 КзБ	-
level_archives	★ Обзор Структура Поиск Вставить Очистить Удалить	15	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
management	★ Обзор Структура Поиск Вставить Очистить Удалить	1 745	InnoDB	utf8mb3_unicode_ci	176.0 КзБ	-
manufacturing_method	★ Обзор Структура Поиск Вставить Очистить Удалить	8	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-

Рис. 2.1. Структура базы данных АФУ(ч.1)

material_records	★ Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
racks	★ Обзор Структура Поиск Вставить Очистить Удалить	16	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
repositories	★ Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
roles	★ Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8mb4_unicode_ci	16.0 КзБ	-
second_language_records	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	latin1_swedish_ci	16.0 КзБ	-
shelves	★ Обзор Структура Поиск Вставить Очистить Удалить	96	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-
sign_untruly	★ Обзор Структура Поиск Вставить Очистить Удалить	7	InnoDB	latin1_swedish_ci	16.0 КзБ	-
stat	★ Обзор Структура Поиск Вставить Очистить Удалить	85	InnoDB	utf16_unicode_ci	32.0 КзБ	-
status_records	★ Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8mb3_unicode_ci	16.0 КзБ	-

Рис. 2.2. Структура базы данных АФУ(ч.2)

um_content_field_allegations	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~112 809	InnoDB	utf8mb3_unicode_ci	5.5 MB	-
um_content_field_foto	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	6 246	InnoDB	utf8mb3_unicode_ci	1.5 MB	-
um_content_type_kartka	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~81 088	InnoDB	utf8mb3_unicode_ci	47.6 MB	-
um_files	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	5 518	InnoDB	latin1_swedish_ci	1.5 MB	-
um_nationality	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	59	InnoDB	utf8mb3_unicode_ci	16.0 KKB	-
um_node	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~57 124	InnoDB	utf8mb3_unicode_ci	7.5 MB	-
um_og_ancestry	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~57 566	InnoDB	latin1_swedish_ci	2.5 MB	-
um_term_data	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	620	InnoDB	utf8mb3_unicode_ci	64.0 KKB	-
unique_records	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	0	InnoDB	utf8mb3_unicode_ci	16.0 KKB	-
users	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	17	InnoDB	utf8mb3_general_ci	16.0 KKB	-
verification	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	10	InnoDB	utf8mb3_unicode_ci	16.0 KKB	-
view_unique	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	440	InnoDB	utf8mb3_unicode_ci	1.5 MB	-
view_unique_en	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	280	InnoDB	utf8mb3_unicode_ci	192.0 KKB	-
z_fonds_names_test	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~395 382	InnoDB	utf8mb3_unicode_ci	106.6 MB	-
z_fonds_parts_test	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~238 535	InnoDB	utf8mb3_unicode_ci	96.6 MB	-
z_fonds_test	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~257 514	InnoDB	utf8mb3_unicode_ci	26.6 MB	-
z_test	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	~152 328	InnoDB	latin1_swedish_ci	22.5 MB	-
z_test2	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить	2 416	InnoDB	latin1_swedish_ci	320.0 KKB	-

Рис. 2.3. Структура бази даних АФУ(ч.3)

На рисунках 2.1 – 2.3 зображена структура бази даних архівних фондів України та всі таблиці, які входять до цієї бази. Для модернізації та розробки нової бази даних було проаналізовано такі таблиці, як:

- um_content_field_allegations;
- um_content_field_foto;
- um_content_type_kartka;
- um_files;
- um_nationality;
- um_node;
- um_og_ancestry;
- um_term_data.

Таблиця «um_content_field_allegations» відповідає за зберігання інформації про звинувачення репресованої особи.

Таблиця «um_content_field_foto» відповідає за зберігання інформації про шлях на накопичувачі до фотокарток зв'язаних з репресованою особою.

Таблиця «um_content_type_kartka» відповідає за зберігання основної інформації про репресовану особу. В ній знаходиться така інформація, як дати

народження та смерті, місце проживання, рід занять, інформація про звинуваченого, тощо.

Таблиця «um_files» відповідає за зберігання інформації про файли, які зв'язані з репресованою особою, їх назвою, розміром та типом файлу.

Таблиця «um_nationality» відповідає за зберігання інформації про національність репресованої особи.

Таблиця «um_node» відповідає за зберігання інформації про установу чи особу, яка надала інформацію про репресовану особу.

Таблиця «um_og_ancestry» відповідає за зв'язок між таблицями.

Таблиця «um_term_data» відповідає за зберігання інформації про вирок репресованої особи.

Проаналізувавши вище вказані таблиці було сформовано представлення нової модернізованої бази даних, в якій таблиці будуть називатися більш зрозуміло, інформація буде розділена по таблицям за типом та важливістю.

2.2 Розробка структури нової бази даних

Розробка структури та зв'язки бази даних було вирішено виконувати у веб-ресурсі «nosqldbm», завдяки простому інтерфейсу та малим системним вимогам.

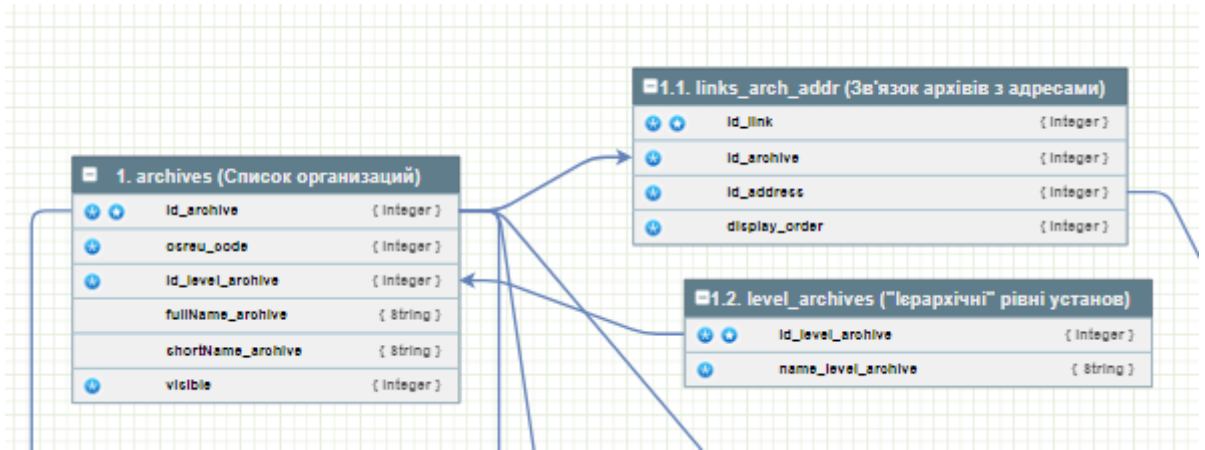


Рис. 2.5. Зображення таблиць "archives"

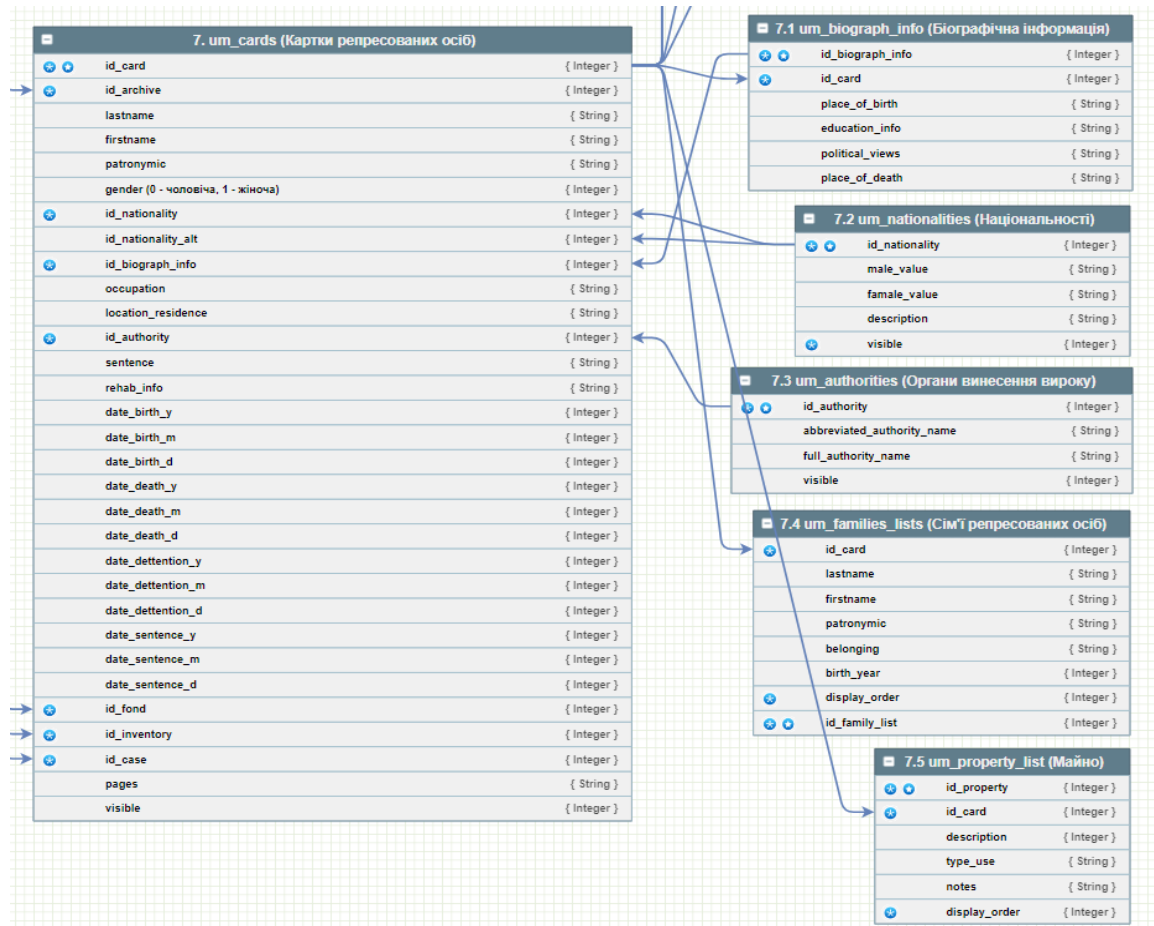


Рис. 2.9. Зображення таблиць "um_cards"(ч.1)

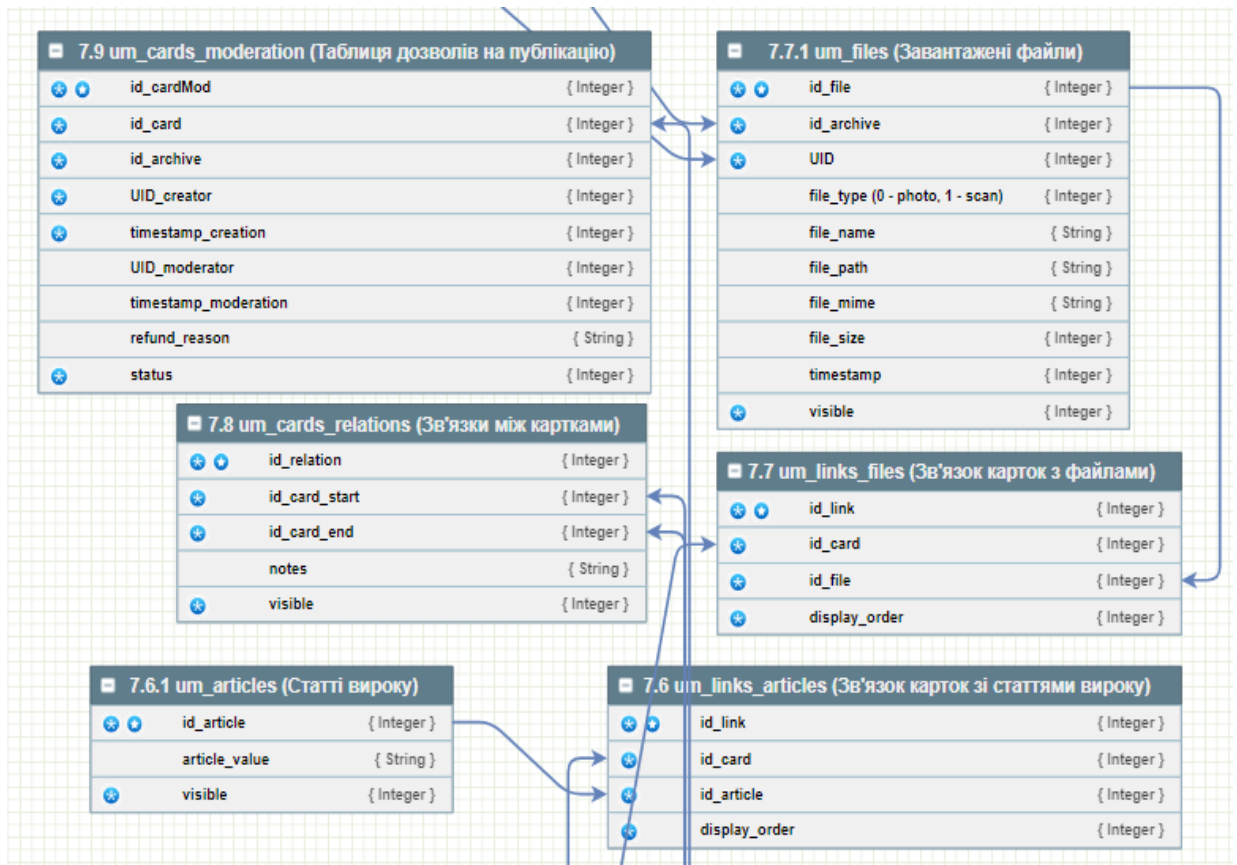


Рис. 2.10. Зображення таблиць “um_cards”(ч.2)

На рисунках 2.9 та 2.10 показано таблиці з інформацією про репресовану особу:

- 7 um_cards (Картки репресованих);
- 7.1 um_biographical_data (Біографічні дані);
- 7.2 um_nationalities (Національності);
- 7.3 um_authorities (Органи, який виносили вироки);
- 7.4 um_kinsman (Родичі репресованих);
- 7.5 um_property (Майно репресованих);
- 7.6 um_articles (Статті вироку);
- 7.7 um_links_cards_articles (Зв'язок карток зі статтями вироку).

2.3 Створення модернізованої бази даних

Для створення бази даних використовується веб-додаток “phpMyAdmin”. phpMyAdmin — це безкоштовний програмний інструмент, написаний на PHP, призначений для адміністрування MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій на MySQL і MariaDB. Часто використовувані операції (керування базами даних, таблицями, стовпцями, зв’язками, індексами, користувачами, дозволами тощо) можна виконувати через інтерфейс користувача, при цьому ви все ще маєте можливість безпосередньо виконувати будь-який оператор SQL.

Створення таблиць зображено на рисунках 2.11 та 2.12.

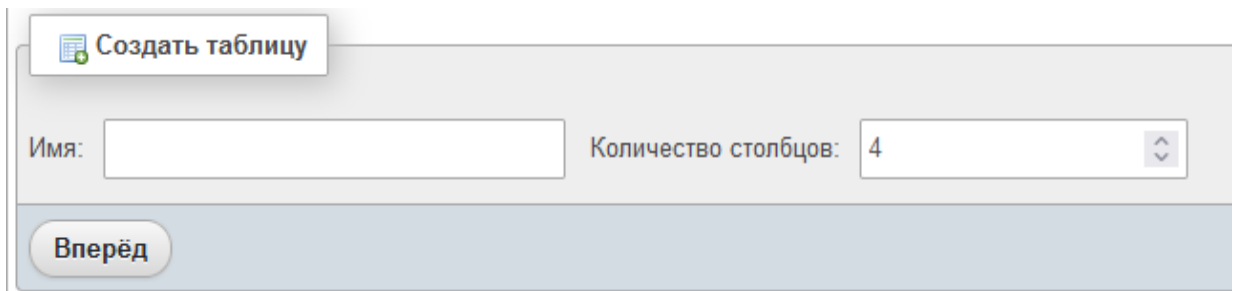


Рис. 2.11. Зображення створення нової таблиці(ч.1)

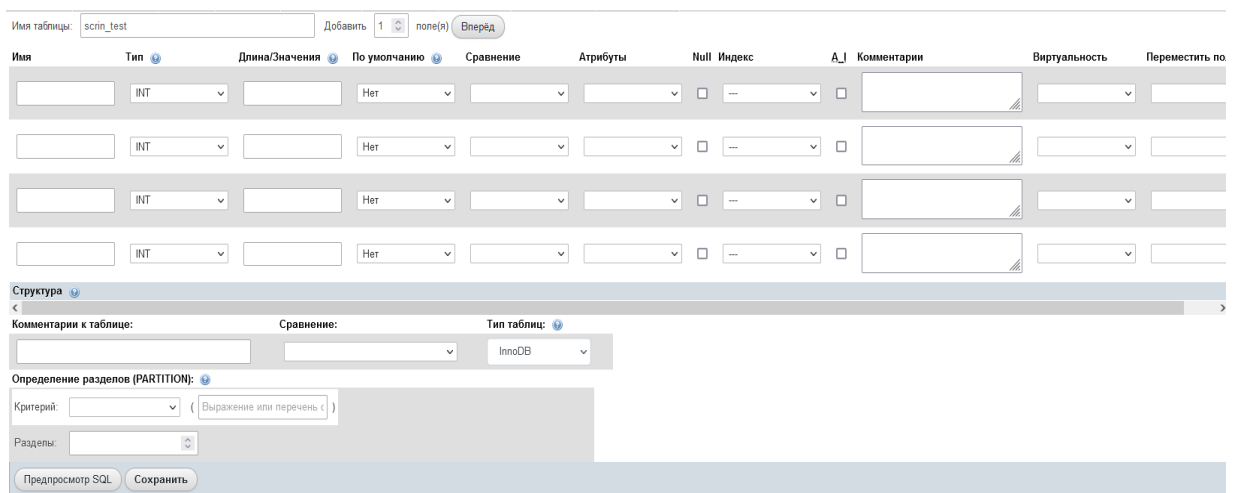


Рис. 2.12. Зображення створення нової таблиці(ч.2)

При додаванні нової таблиці до бази даних, потрібно спочатку ввести назву для таблиці, потім задати кількість стовпців і натиснути на кнопку «Далі»

(зображено на рисунку 2.11 (ч.1)). Наступним кроком при додаванні нової таблиці є називання стовпців, визначення та призначення їх типів, задання максимальної кількості знаків вмістимих в комірку. Далі призначення атрибутів для стовпців, індексів та коментарів, для простішого розуміння призначення стовпців при роботі з базою даних в майбутньому.

Таблиця 2.1

Вимоги заповнення таблиці «archive»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
1.1	id_archive	Number	1	Унікальний цифровий код (ідентифікатор) архівної установи	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
1.2	osreu_code	Number	1	Цифровий код (номер) архівної установи в Єдиному державному реєстрі підприємств та організацій України (ЄДРПОУ)	Вводиться вручну
1.3	id_level_archive	Number	1	Унікальний цифровий код (ідентифікатор) рівня архівної установи	Вибір із списку, що автоматично формується із даних таблиці 1.2 level_archives (Рівні архівних установ) цієї бази даних
1.4	fullName_archive	Text	1	Повне найменування архівної установи	Вводиться вручну

Продовження таблиці 2.1

Вимоги заповнення таблиці «archive»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
1.5	shortName_archive	Text	0	Скорочена назва архівної установи. У разі офіційного застосування	Вводиться вручну
1.6	visible	Number	1	Статус відображення: 1 – відобразити, 0 – приховати	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Таблиця 2.2

Вимоги заповнення таблиці «»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.1	id_card	Number	1	Унікальний цифровий код (ідентифікатор) картки репресованого	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.2	id_archive	Number	1	Унікальний цифровий код (ідентифікатор) архівної установи	Вибір із списку, що автоматично формується із даних таблиці 1 archives (Архівні установи)

Продовження таблиці 2.2

Вимоги заповнення таблиці «um_cards»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.3	id_fond	Number	1	Унікальний цифровий код (ідентифікатор) фонду	Вибір із списку, що автоматично формується із даних таблиці 2 fonds (Фонди)
7.4	id_inventory	Number	1	Унікальний цифровий код (ідентифікатор) опису	Вибір із списку, що автоматично формується із даних таблиці 3 inventories (Описи)
7.5	id_case	Number	1	Унікальний цифровий код (ідентифікатор) справи	Вибір із списку, що автоматично формується із даних таблиці 4 cases (Справи)
7.6	repressed_lastname	Text	1	Прізвище репресованого	Вводиться вручну
7.7	repressed_firstname	Text	1	Ім'я репресованого	Вводиться вручну
	repressed_patronymic	Text	0	По-батькові репресованого	Вводиться вручну
7.8	repressed_gender	Number	1	Стать репресованого: 0 – чоловіча, 1 – жіноча	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Продовження таблиці 2.2

Вимоги заповнення таблиці «um_cards»

Ін-декс	Назва поля	Тип даних	Обов'яз-ковість	Описування	Правила заповнення
7.10	id_nationality_alternative	Number	0	Унікальний цифровий код (ідентифікатор) альтернативної національності	
7.11	id_biograph_info	Number	0	Унікальний цифровий код (ідентифікатор) біографічних даних	Вибір із списку, що автоматично формується із даних таблиці 7.1 um_biograph_info
7.12	location_detention	Text	0	Місце проживання на момент арешту	Вводиться вручну
7.13	date_detention_y	Number	0	Рік арешту	Вводиться вручну
7.14	date_detention_d	Number	0	День арешту	Вводиться вручну
7.15	date_detention_m	Number	0	Місяць арешту	Вводиться вручну
7.16	id_authority	Number	0	Унікальний цифровий код (ідентифікатор) органу винесення вироку	Вибір із списку, що автоматично формується із даних таблиці 7.3 um_authorities
7.17	date_sentence_y	Number	0	Рік вироку	Вводиться вручну

Продовження таблиці 2.2

Вимоги заповнення таблиці «um_cards»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.18	date_sentence_m	Number	0	Місяць вироку	Вводиться вручну
7.19	date_sentence_d	Number	0	День вироку	Вводиться вручну
7.20	sentence	Text	0	Вирок	Вводиться вручну
7.21	location_residence	Text	0	Місце реабілітації	Вводиться вручну
7.22	rehab_info	Text	0	Інформація про реабілітацію	Вводиться вручну
7.23	id_user	Number	1	Ідентифікатор користувача, який створив картку	Вибір із списку, що автоматично формується із даних таблиці 6 users (Користувачі)
7.24	visible	Number	1	Статус відображення: 1 – відобразити, 0 – приховати	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Таблиця 2.3

Вимоги заповнення таблиці «um_biographical_data»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.1.1	id_biographical_data	Number	1	Унікальний цифровий код (ідентифікатор) біографічних даних	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.1.2	id_card	Number	0	Унікальний цифровий код (ідентифікатор) картки репресованого	Вибір із списку, що автоматично формується із даних таблиці 7 um_cards (Картки репресованих)
7.1.4	place_of_birth	Text	0	Місце народження	Вводиться вручну
7.1.5	date_birth_y	Number	0	Рік народження	Вводиться вручну
7.1.6	date_birth_m	Number	0	Місяць народження	Вводиться вручну
7.1.7	date_birth_d	Number	0	День народження	Вводиться вручну
7.1.8	education_info	Text	0	Інформація про освіту	Вводиться вручну
7.1.9	political_views	Text	0	Політичні погляди	Вводиться вручну
7.1.10	occupation	Text	0	Рід заняття	Вводиться вручну
7.1.11	place_of_death	Text	0	Місце смерті	Вводиться вручну
7.1.12	date_death_y	Number	0	Рік смерті	Вводиться вручну
7.1.13	date_death_m	Number	0	Місяць смерті	Вводиться вручну
7.1.14	date_death_d	Number	0	День смерті	Вводиться вручну

Таблиця 2.4

Вимоги заповнення таблиці «um_nationality»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.2.1	id_nationality	Number	1	Унікальний цифровий код (ідентифікатор) національності	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.2.2	male_value	Text	1	Національність чоловічого роду	Вводиться вручну
7.2.3	famale_value	Text	1	Національність жіночого роду	Вводиться вручну
7.2.4	description	Text	0	Описання. Зазначається за потребою	Вводиться вручну
7.2.5	visible	Number	1	Статус відображення: 1 – відобразити, 0 – приховати	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Таблиця 2.5

Вимоги заповнення таблиці «um_authority»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.3.1	id_authority	Number	1	Унікальний цифровий код (ідентифікатор) органу, який виносив вироки	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.3.2	shortName_authority	Text	1	Повне найменування органу, який виносив вироки	Вводиться вручну
7.3.3	fullName_authority	Text	0	Скорочена назва органу, який виносив вироки. У разі офіційного застосування	Вводиться вручну
7.3.4	visible	Number	1	Статус відображення: 1 – відобразити, 0 – приховати	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Таблиця 2.6

Вимоги заповнення таблиці «um_kinsman»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.4.1	id_kinsman	Number	1	Унікальний цифровий код (ідентифікатор) коду родича репресованого	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.4.2	id_card	Number	1	Унікальний цифровий код (ідентифікатор) картки репресованого	Вибір із списку, що автоматично формується із даних таблиці 7 um_cards (Картки репресованих)
7.4.3	kinsman_lastname	Text	0	Прізвище родича	Вводиться вручну
7.4.4	kinsman_firstname	Text	0	Ім'я родича	Вводиться вручну
7.4.5	kinsman_patronymic	Text	0	По-батькові родича	Вводиться вручну
7.4.6	kinsman_birth_year	Number	0	Рік народження родича репресованого	Вводиться вручну
7.4.7	kinsman_belonging	Text	1	Рідство. Приклад: син, брат і т.д.	Вводиться вручну
7.4.8	id_card_kinsman	Number	0	Унікальний цифровий код (ідентифікатор) картки репресованого родича.	Вибір із списку, що автоматично формується із даних таблиці 7 um_cards (Картки репресованих)
7.4.9	display_order	Number	1	Порядок відображення ролей.	Вводиться вручну

Таблиця 2.7

Вимоги заповнення таблиці «um_property»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.5.1	id_property	Number	1	Унікальний цифровий код (ідентифікатор) код одиниці майна	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.5.2	id_card	Number	1	Унікальний цифровий код (ідентифікатор) картки репресованого	Вибір із списку, що автоматично формується із даних таблиці 7 um_cards (Картки репресованих)
7.5.3	property_description	Text	1	Описування майна	Вводиться вручну
7.5.4	display_order	Number	1	Порядок відображення ролей. Число, починаючи від нуля	Вводиться вручну

Таблиця 2.8

Вимоги заповнення таблиці «um_article»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.6.1.1	id_article	Number	1	Унікальний цифровий код (ідентифікатор) статті	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.6.1.2	article_label	Text	1	Позначення статті. Приклад: 54-1а КК УРСР	Вводиться вручну
7.6.1.3	article_text	Text	0	Зміст статті. Зазначається у разі наявності та відповідно до чинного на час винесення вироку Карного кодексу	Вводиться вручну
7.6.1.4	visible	Number	1	Статус відображення: 1 – відобразити, 0 – приховати	Вводиться вручну за допомогою відповідного елемента інтерфейсу, наприклад: прапорець

Таблиця 2.9

Вимоги заповнення таблиці «um_links_cards_articles»

Індекс	Назва поля	Тип даних	Обов'язковість	Описування	Правила заповнення
7.6.1	id_link	Number	1	Унікальний цифровий код (ідентифікатор) зв'язку карток зі статтями вироку	Наскрізна нумерація. Створюється автоматично при додаванні нового запису в цю таблицю
7.6.2	id_card	Number	1	Унікальний цифровий код (ідентифікатор) картки репресованого	Вибір із списку, що автоматично формується із даних таблиці 7 um_cards (Картки репресованих)
7.6.3	id_article	Number	0	Унікальний цифровий код (ідентифікатор) статті	Вибір із списку, що автоматично формується із даних таблиці 7.6 um_articles
7.6.4	display_order	Number	1	Порядок відображення ролей. Число, починаючи від нуля	Вводиться вручну

2.4 Аналіз веб-форми

В ході аналізу різних веб-форм було вирішено використовувати основною мовою програмування PHP. Також для коректної роботи форми наповнення бази даних використовується HTML, CSS та JavaScript мови програмування.

HTML відповідає за розмітку сторінки на якій буде реалізована веб-форма для наповнення бази даних. За допомогою HTML буде створено основна частина веб-форми:

- поля вводу;
- випадаючі списки;
- кнопки керування.

Поля вводу служать для внесення даних вручну користувачем так, як додавання таких даних неможливо реалізувати за рахунок випадаючих списків.

Випадаючі списки використовують для спрощення додавання інформації про особу, такі дані дуже схожі при їх додаванні:

- день;
- місяць;
- дані які мають більш структуровану будову і їх можливо підтягувати в списки за допомогою запитів в базу даних.

Кнопки керування використовуються для наступних цілей:

- додання особливого поля, в якому обирається альтернативна національність репресованої особи(у випадку, якщо за своє життя особа змінювала свою національність);
- додавання особливих полів вводу даних, які при внесенні до бази даних будуть записані в одну комірку тому, що мають однакову суть але відрізняються за значенням і їх може бути більше ніж одна у особи, про яку вноситься картка до БД («стаття», «одиниця майна»);
- відкриття модального вікна для більш зручного додавання інформації;

- додання картки до БД (у разі завершення вводу даних та введення всіх обов'язкових полів).

Мова програмування CSS використовується для стилізації форми, тобто для коректного відображення форми та приємнішої роботи користувача з нею. Для простішого та більш динамічного інтерфейсу було обрано для використання бібліотеки Bootstrap. Данна бібліотека дає змогу налаштувати та стилізувати сторінку з веб-формою вже перевіреними способами, завдяки яким на даний момент працює дуже багато сайтів.

JavaScript використовується для взаємодії користувача та веб-форми, під час якої користувач може змінювати структуру веб-форми, а також її вигляд.

2.5 Розробка інтерфейсу бази даних

Веб-форма буде виконана на одній сторінці, так як полів для вводу інформації небагато і однієї сторінки достатньо для повного відображення та зручності користування.

На цій сторінці буде виконано розділення форми на підрозділи, це дасть можливість компонувати інформацію по значимості та типу. Таким чином при введенні інформації про репресовану особу, користувачеві не потрібно буде шукати по всій формі потрібне поле для вводу, а дивитися в спеціальний розділ.

У верхній частині веб-сторінки(header) заходиться найменування форми, показано на рисунку 2.13.

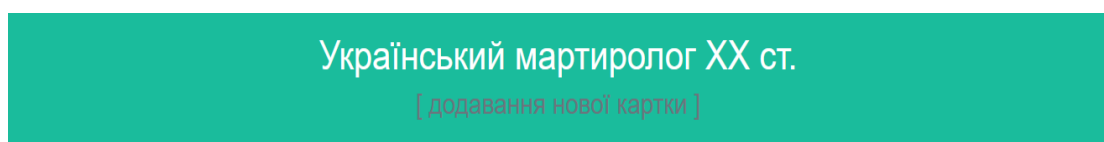


Рис. 2.13. Header веб-форми

Всі поля для вводу знаходять в body. Першим розділом для вводу даних буде загальна інформація.

Загальна інформація

Прізвище	Введіть прізвище	Ім'я	Введіть ім'я	По-батькові	Введіть по-батькові	Стать:	не обрано	▼
Національність:	не обрано	+						

Рис 2.14. Розділ для вводу загальної інформації

Загальна інформація

Прізвище	Введіть прізвище	Ім'я	Введіть ім'я	По-батькові	Введіть по-батькові	Стать:	не обрано	▼
Національність:	не обрано	▼	Національність #2:	не обрано	▼	×		

Рис 2.15. Розділ для вводу загальної інформації з додатковим полем

На рисунку 2.14 зображено перший розділ для вводу інформації, в якому знаходяться такі поля:

- Прізвище;
- Ім'я;
- По-батькові;
- Стать;
- Національність.

Поля для вводу «Прізвище», «Ім'я», «По-батькові» - звичайні поля для вводу, при завершенні додавання інформації до форм, інформація з них записується у відповідні комірки бази даних.

«Стать» та «Національність» - випадючі списки, в яких можна обрати інформацію яка підтягується з бази даних.

На рисунку 2.15 зображено той же самий розділ для вводу загальної інформації але з доданим випадючим списком. Він служить для вибору

альтернативної національності, який додається до форми при натисканні на «+». При натисканні на «x» це поле видаляється з форми.

Бібліографічні дані

Дата народження:	День	▼	Місяць	▼	Рік	Місце народження	Введіть місце народження
Освіта	Введіть інформацію про освіту						
Рід занять	Введіть рід занять						
Політичні погляди	Введіть політичні погляди						
Дата смерті:	День	▼	Місяць	▼	Рік	Місце смерті	Введіть місце смерті

Рис. 2.15. Розділ для вводу біографічних даних

На рисунку 2.15 зображено розділ вводу біографічних даних про репресовану особу, в якому знаходяться такі поля:

- Дата народження;
- Місце народження;
- Освіта;
- Рід занять;
- Політичні погляди;
- Дата смерті;
- Місце смерті.

Поля для вводу «Місце народження», «Освіта», «Рід занять», «Політичні погляди» «Місце смерті» - звичайні поля для вводу, при завершенні додавання інформації до форм, інформація з них записується у відповідні комірки бази даних.

Поля для вводу «Дата народження», «Дата смерті» - знаходять в собі два випадючі списки, з вибором дня(з 1 по 31 число) та вибором місяця(з 1 по 12 місяць), а також звичайне поле для вводу, в яке потрібно вручну вписати рік.

Інформація про вирок

Дата арешту:	День	▼	Місяць	▼	Рік	Місце проживання на момент арешту	Введіть місце проживання	
Стаття #1:	не обрано						▼	+
Дата вироку:	День	▼	Місяць	▼	Рік	Орган винесення вироку:	не обрано	
Вирок:	Введіть інформацію про вирок							

Рис. 2.16. Розділ для вводу інформації про вирок

Інформація про вирок

Дата арешту:	День	▼	Місяць	▼	Рік	Місце проживання на момент арешту	Введіть місце проживання	
Стаття #1:	не обрано						▼	+
Стаття #2:	не обрано						▼	×
Дата вироку:	День	▼	Місяць	▼	Рік	Орган винесення вироку:	не обрано	
Вирок:	Введіть інформацію про вирок							

Рис. 2.17. Розділ для вводу інформації про вирок з додатковим полем

На рисунку 2.16 зображено розділ для вводу інформації про вирок, в якому знаходяться такі поля:

- Дата арешту;
- Місце проживання на момент арешту;
- Стаття;
- Дата вироку;
- Орган винесення вироку;
- Вирок.

Поля для вводу «Місце проживання на момент арешту», «Орган винесення вироку», «Вирок» - звичайні поля для вводу, при завершенні додавання інформації до форм, інформація з них записується у відповідні комірочки бази даних.

Поля для вводу «Дата арешту», «Дата вироку» - знаходять в собі два випадуючі списки, з вибором дня(з 1 по 31 число) та вибором місяця(з 1 по 12 місяць), а також звичайне поле для вводу, в яке потрібно вручну вписати рік.

Поле вводу «Стаття» - випадуючий список, що формується із записів, які вже є в базі даних.

На рисунку 2.17 зображено розділ для вводу інформації про вирок з додатковим полем «Стаття», доданого для можливості додавання інформації про більше ніж одну статтю. Максимальна кількість таких полів складає 5, дане число було отримано з огляду карток в старій базі даних, в яких було зазначено максимальна кількість статей для однієї особи – 5 статей. Це поле можна додати до форми при натисканні на «+». При натисканні на «x» це поле видаляється з форми.

Інформація про реабілітацію

Місце реабілітації	Введіть інформацію про місце реабілітації
Відомості про реабілітацію	Введіть відомості про реабілітацію

Рис. 2.18. Розділ для вводу інформації про реабілітацію

На рисунку 2.18 зображено розділ для вводу інформації про реабілітацію, до якого входять такі поля:

- Місце реабілітації;
- Відомості про реабілітацію.

Поля для вводу «Місце реабілітації», «Відомості про реабілітацію» - звичайні поля для вводу, при завершенні додавання інформації до форм, інформація з них записується у відповідні комірки бази даних.

В поле «Відомості про реабілітацію» вводиться інформація про надлишкові відомості з місця реабілітації, поле не обов'язкове для заповнення.

Інформація про майно

Одиниця майна #1	Введіть відомості про одиницю майна	+
------------------	-------------------------------------	---

Рис. 2.19. Розділ для вводу інформації про майно

Інформація про майно

Одиниця майна #1	Введіть відомості про одиницю майна	+
Одиниця майна #2:	Введіть відомості про одиницю майна	×

Рис. 2.20. Розділ для вводу інформації про майно з додатковим полем

На рисунку 2.19 зображено розділ для вводу інформації про майно репресованої особи, який містить в собі лише одне поле для вводу: «Одиниця майна». В дане поле вводиться інформація про тип, цінність та технічні характеристики (якщо мова йде про нерухомість) майна, яке належало особі.

На рисунку 2.20 зображено розділ для вводу інформації про майно з додатковим полем, доданого для можливості додавання інформації про більш ніж одну одиницю майна. Максимальна кількість таких полів складає 5, дане число було отримано з огляду карток в старій базі даних, в яких було зазначено максимальна кількість статей для однієї особи – 5 одиниць майна. Це поле можна додати до форми при натисканні на «+». При натисканні на «x» це поле видаляється з форми.

Інформація про родичів

Пошук в базі даних:	Пошук родича в базі даних
---------------------	---------------------------

Додати інформацію про родича вручну

Рис. 2.21. Розділ для вводу інформації про родичів

На рисунку 2.21 зображено розділ для вводу інформації про родичів репресованої особи, до якого належать такі елементи:

- Пошук в базі даних;
- Додати інформацію про родича вручну.

Поле вводу «Пошук в базі даних» - поле, яке служить для пошуку інформації про родича репресованої особи в базі даних у разі, якщо інформація про нього була раніше додана в базу.

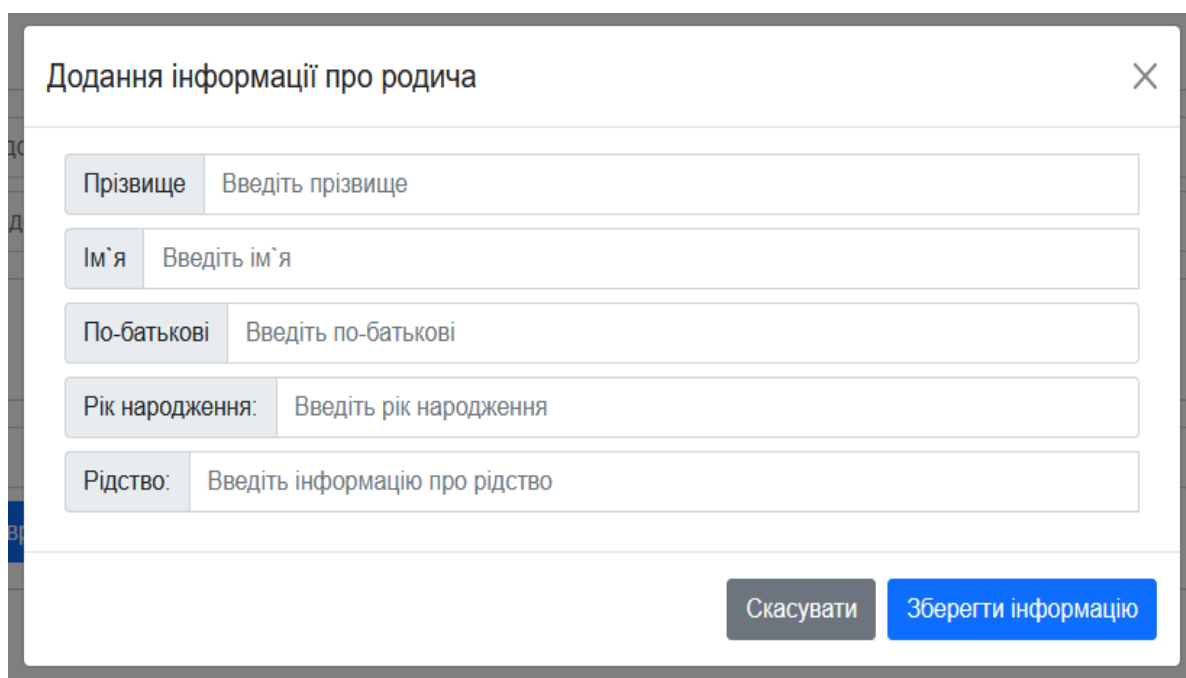


Рис. 2.22. Модальне вікно для вводу інформації про родича

На рисунку 2.22 зображено модальне вікно для вводу інформації про родича, яке відкривається при натисканні на кнопку «Додати інформацію про родича вручну» (показано на рис. 2.21.), в якому знаходяться такі поля:

- Прізвище;
- Ім'я;
- По-батькові;
- Рік народження;

- Рідство.

Поля для вводу «Прізвище», «Ім'я», «По-батькові», «Рік народження», «Рідство» - звичайні поля для вводу, при завершенні додавання інформації до форм, інформація з них записується у відповідні комірки бази даних. Також у формі присутні кнопки «Скасувати», «x» - кнопки, які відповідають за відміну додавання інформації про родича та закриття модального вікна для продовження роботи з формою. Кнопка «Зберегти інформацію» відповідає за збереження інформації про родича у випадку заповнення полів.

Рис. 2.23 Валідоване модальне вікно

Також кнопка «Зберегти інформацію» відповідає за валідацію модального вікна, тобто перевіряє чи заповнені обов'язкові поля вводу (показано на рис. 2.23).

Рис. 2.24 Розділ для вводу інформації про родичів з доданою інформацією

В разі правильного заповнення полів вводу в модальному вікні та натисканні на кнопку «Зберегти інформацію», модальне вікно закривається, а інформація введена в ньому зберігається в спеціальному полі, що формується під кнопкою «Додати інформацію про родича вручну» (показано на рис. 2.24.). Дане поле створюється для перевірки правильності заповнення інформації про родича і в разі потреби редагування, чи видалення цієї інформації, за допомогою кнопок з правого боку від поля з інформацією.

При натисканні на кнопку редагування відкривається модальне вікно з інформацією, яку було записано раніше і надається можливість редагування цієї інформації.

Архівні відомості

Архівна установа: не обрано

№ фонду Введіть номер фонду

№ опису Введіть номер опису

№ справи Введіть номер справи

Рис. 2.25. Розділ для вводу даних про архівні відомості

На рисунку 2.25 зображено розділ для вводу даних про архівні відомості, до якого належать такі поля:

- Архівна установа;
- № фонду;
- № опису;
- № справи.

Поле «Архівна установа» - випадаючий список, який надає можливість обрати назву архівної установи (що записані в базі даних) для ідентифікації картки про репресовану особу.

Поля «№ фонду», «№ опису», «№ справи» - звичайні поля для вводу, при завершенні додавання інформації до форми, інформація з них записується у відповідні комірки бази даних.



Рис. 2.26. Кнопка додавання картки в БД

Після завершення вводу даних у форму, потрібно натиснути на кнопку «+ Додати картку до БД» (показано на рис. 2.26), щоб інформація обробилась і записалась в базу даних.

2.6 Розробка функціональної частини веб-форми

Для коректного відпрацювання веб-форми наповнення бази даних, було вирішено використовувати скрипти, написані мовою програмування JavaScript. Також було використано бібліотеку jQuery. jQuery - це швидка, невелика і багатофункціональна бібліотека JavaScript. Вона значно спрощує такі речі, як обхід і маніпуляції з HTML, обробка подій, анімація і Ajax за допомогою простого у використанні API, який працює у безлічі браузерів. Завдяки поєднанню універсальності і розширюваності jQuery змінив спосіб написання JavaScript мільйонами людей.

Також дає можливість маніпулювання DOM (Document Object Model), в якому є декілька методів, як це зробити. Простіше кажучи, він дозволяє вставляти і/або видаляти елементи DOM на HTML- сторінці, а також спрощує перенесення рядків. Створення анімації також спрощене за допомогою jQuery. Як і в приведеному вище фрагменті коду про анімацію, він покритий декількома рядками коду, все що вам треба зробити, це вставити змінні.

```

<script type="text/javascript">
$( '#plus_1' ).click(function(){
// let nac_block = $('#nac');
// nac_block.attr('id', 'nac_alt');
// console.log(nac_block);

$('#nac').after(
'<div class="col-3" id="nac_alt">' +
'<div class="input-group">' +
'<label for="nationality-2" class="input-group-text" style="text-align: right;">Національність #2:</label>' +
'<select class="form-select" name="nationality-2">' +
'<option value="">' +
'</option>' +
'</select>' +
'</div>' +
'</div>' +
'<div class="col">' +
'<span class="d-inline-block" tabindex="0" data-toggle="tooltip" title="Прибрати поле">' +
'<button type="button" class="btn btn-danger minus_1">' +
'<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-x-lg" viewBox="0 0 16 16">' +
'<path fill-rule="evenodd" clip-rule="evenodd" d="M3.854 2.146a.5.5 0 0 1 0 .708l-11 11a.5.5 0 0 1-.708-.708l-11-11a.5.5 0 0 1.708 0z" fill="#000"/>' +
'<path fill-rule="evenodd" clip-rule="evenodd" d="M2.146 2.146a.5.5 0 0 0 0 .708l11 11a.5.5 0 0 0-.708-.708l-11-11a.5.5 0 0 0.708 0z" fill="#000"/>' +
'</svg>' +
'</button>' +
'</span>' +
'</div>'
);
});

$(document).on('click', '.plus_1', function(){
$('#plus_1').attr('style', 'display:none');
});

$(document).on('click', '.minus_1', function(){
$('#minus_1').closest('div').remove();
$('#nac_alt').remove();
$('#plus_1').removeAttr('style');
});
</script>

```

Рис 2.27. Кодова частина обробки національності

Створення додаткового випадаючого списку з альтернативною національністю (як показано на рис. 2.15.) показано на рисунку 2.27, на якому зображено кодову частину обробки натискання на кнопку додавання поля.

При натискання на кнопку додання поля виконується створення нового DOM-елемента з копією першого поля «Національність», та створення кнопки для видалення цього поля (в разі його непотрібності).

Також під час створення нового поля, йому присвоюється спеціальний ідентифікатор, щоб під час обробки та запису в базу даних поля національності та альтернативної національності, записувалися в правильні комірочки.

```

?php
// Витягаємо із бази перелік статей
$query = "SELECT id_article,article_label FROM `test_um_articles` WHERE `visible` = 1 ORDER BY article_label+0 ASC, article_label ASC";
$result = $pdo->query($query); //Виконання запити
$rows = $result->fetchAll(PDO::FETCH_ASSOC); //Збереження всіх результатів запити в масиві

```

Рис. 2.28. Кодова частина списку переліку статей

На рисунку 2.28 зображено кодову частину за допомогою якої виконується формування списку зі статтями, для простішого вводу даних про репресовану особу користувачем. Цей процес реалізується завдяки РНР-запиту до бази даних та перебору даних в ній стосовно статей.

```
<?php
foreach($rows as $row_num => $row_value) {
    if(!$row_num) {
        echo '<option selected disabled value="">не обрано</option>';
    }
    echo '<option value="" . $row_value['id_article'] . "'>' . $row_value['article_label'] . '</option>';
}
?>
```

Рис. 2.29. Кодова частина відображення списку статей

На рисунку 2.29 зображена кодова частина, завдяки якій відбувається відображення випадючого списку зі всіма доступними статтями. Інформація виводиться у вигляді масиву із комірок «article_label».

```
<script type="text/javascript">
var art_id = 1;
$('.article_add').click(function(){
    let last_elem = $('#div[id*=article]:last');
    art_id += 1;

    last_elem.after(
        '<div class="row mb-2" id="article_' + art_id + '">' +
        '<div class="col">' +
        '<div class="input-group">' +
        '<label for="article_' + art_id + '" class="input-group-text" style="text-align: right;">Стаття #' + art_id + ':</label>' +
        '<select class="form-select" name="article_' + art_id + '">' +
        $('#select[name="article_1"]')[0].innerHTML +
        '</select>' +
        '</div>' +
        '</div>' +
        '<div class="col-art-add">' +
        '<span class="d-inline-block" tabindex="0" data-toggle="tooltip" title="Видалити поле">' +
        '<button type="button" class="btn btn-danger article_del">' +
        '<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-x-lg" viewBox="0 0 16 16">' +
        '<path fill-rule="evenodd" clip-rule="evenodd" d="M13.854 2.146a.5.5 0 0 1 0 .708l-11 11a.5.5 0 0 1-.708.708l-11-11a.5.5 0 0 1-.708-.708l-11-11a.5.5 0 0 1.708-.708l11-11a.5.5 0 0 1.708.708l11 11a.5.5 0 0 1 0 .708z" fill="#000"/>' +
        '</svg>' +
        '</button>' +
        '</span>' +
        '</div>' +
        '</div>' +
    );
});

$('body').on('click', '.article_del', function(e) { // видалення статті
    $(this).closest('div[id*=article]').remove(); // видалення рядку зі статтею
    let all_elems = $('#div[id*=article]'); // шукаємо всі статті
    art_id = all_elems.length; // рихуємо їх
    all_elems.each(function (index, element) { //оновлюємо ID рядків
        $(element).attr('id', 'article_' + (index+1)); // замінюємо ID в DIV
        let tmp1 = $(element).find('label'); // шукаємо тег з назвою рядка
        $(tmp1).attr('for', 'article_' + (index+1));
        $(tmp1).html('Стаття #' + (index+1)); // змінюємо назву
        $(element).find('select').attr('name', 'article_' + (index+1)); //шукаємо и замінюємо ID списку
    });
});
</script>
```

Рис. 2.30. Кодова частина створення додаткового поля зі статтею

Створення нового поля для вводу додаткової статті (показано на рис.2.30.) виконується за схожим принципом, як і поле національності. Нове поле формується під основним полем з ідентичним розміром і розміткою, змінюється лише кнопка справа від поля, замість кнопки «+», формується кнопка «x».

Коли формується нове поле, то йому присвоюється спеціальний ідентифікатор, щоб під час запису до бази даних, інформація додавалась окремо. Також в даному обробнику виконується видалення цих ідентифікаторів, в разі якщо виникла потреба видалити поле зі статей, та замінюються всі інші ідентифікатори, для коректного запису до бази даних.

```
<?php
// Витягаємо із бази перелік органів винесення вироку
$query = "SELECT id_authority,shortName_authority FROM `test_um_authorities` WHERE `visible` = 1 ORDER BY shortName_authority+0 ASC, shortName_authority ASC";
$result = $pdo->query($query); //Виконання запиту
$rows = $result->fetchAll(PDO::FETCH_ASSOC); //Збереження всіх результатів запиту в масиві
?>
```

Рис. 2.31. Кодова частина списку органів винесення вироку

На рисунку 2.31 зображено кодову частину за допомогою якої виконується формування списку з органами винесення вироку, для простішого вводу даних про репресовану особу користувачем. Цей процес реалізується завдяки PHP-запиту до бази даних та перебору даних в ній стосовно цих органів.

```
<?php
foreach($rows as $row_num => $row_value) {
    if(!$row_num) {
        echo '<option selected disabled value="">не обрано</option>';
    }
    echo '<option value="' . $row_value['id_authority'] . '">' . $row_value['shortName_authority'] . '</option>';
}
?>
```

Рис. 2.32. Кодова частина відображення списку органів винесення вироку

На рисунку 2.32 зображена кодова частина, завдяки якій відбувається відображення випадаючого списку зі всіма доступними статтями. Інформація виводиться у вигляді масиву із комірок «shortName_authority».

```

<script type="text/javascript">
var prpt_id = 1;
$('.property_add').click(function(){
let last_elem = $('div[id*=property]:last');
prpt_id += 1;

last_elem.after(
'<div class="row mb-2" id="property_' + prpt_id + '">' +
'<div class="col">' +
'<div class="input-group">' +
'<span class="input-group-text" style="text-align: right;">Одиниця майна #' + prpt_id + '</span>' +
'<input name="property_' + prpt_id + '" type="text" class="form-control" placeholder="Введіть відомості про одиницю майна">' +
'</div>' +
'</div>' +
'<div class="col-prop-add">' +
'<span class="d-inline-block" tabindex="0" data-toggle="tooltip" title="Видалити поле">' +
'<button type="button" class="btn btn-danger property_del">' +
'<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-x-lg" viewBox="0 0 16 16">' +
'<path fill-rule="evenodd" clip-rule="evenodd" d="M3.854 2.146a.5.5 0 0 1 0 .708l-11 11a.5.5 0 0 1-.708-.708l-11-11a.5.5 0 0 1 .708 0z" fill="#000"/>' +
'<path fill-rule="evenodd" clip-rule="evenodd" d="M2.146 2.146a.5.5 0 0 0 0 .708l11 11a.5.5 0 0 0 .708-.708l-11-11a.5.5 0 0 0-.708 0z" fill="#000"/>' +
'</svg>' +
'</button>' +
'</span>' +
'</div>' +
'</div>'
);

$('body').on('click', '.property_del', function(e) { // видалення майна
$(this).closest('div[id*=property_]').remove(); // видалення рядку з майном
let all_prt_elems = $('div[id*=property_]'); // шукаємо всі одиниці майна
prpt_id = all_prt_elems.length; // рхуємо їх
all_prt_elems.each(function (index, element) { //оновлюємо ID рядків
$(element).attr('id', 'property_' + (index+1)); // замінюємо ID в DIV
let tmp1 = $(element).find('span.input-group-text'); // шукаємо тег з назвою рядка
$(tmp1).html('Одиниця майна #' + (index+1)); // змінюємо назву
$(element).find('input').attr('name', 'property_' + (index+1)); //шукаємо і замінюємо ID списку
});
});
});
</script>

```

Рис. 2.33. Кодова частина створення додаткового поля одиниці майна

Створення нового поля для вводу додаткової одиниці майна (показано на рис 2.33.) виконується за схожим принципом, як і поле вводу статей. Нове поле формується під основним полем з ідентичним розміром і розміткою, змінюється лише кнопка справа від поля, замість кнопки «+», формується копка «x».

Коли формується нове поле, то йому присвоюється спеціальний ідентифікатор, щоб під час запису до бази даних, інформація додавалась окремо. Також в даному обробнику виконується видалення цих ідентифікаторів, в разі якщо виникла потреба видалити поле з одиницею майна, та замінюються всі інші ідентифікатори, для коректного запису до бази даних.

```
let modal_info = {};  
$('div[id*="kins_info"] input').each(function (num, element) {  
  let element_name = $(element).attr('name');  
  if($(element).val().trim() !== '') {  
    modal_info[element_name] = $(element).val().trim();  
  }  
  element.value = '';  
});  
let JSON_out = toJSON(modal_info);  
$('.modal').modal('hide');  
  
let last_elem = '';  
if (edit_kins_line !== '') {  
  last_elem = $('div[id*="" + edit_kins_line + "']');  
} else {  
  last_elem = $('div[id*="kins_json_info"]:last');  
  if (!last_elem.length) {  
    last_elem = $('div[id*=add_info_kins]');  
  }  
}
```

Рис. 2.34. Кодова частина модального вікна

На рисунку 2.34 зображена кодова частина обробки модального вікна. В ньому показано запис інформації введеної в модальному вікні та подальшому форматуванні цієї інформації в формат даних JSON.

Формат даних JSON використовується для скорочення кодової частини обробника та збільшення швидкості відпрацювання цього обробника.

Створення кнопок керування полем з інформацією про родича, виконуються водночас зі створення самого поля. Під час створення як полю, так і кнопкам присвоюються спеціальні ідентифікатори. Вони служать для того щоб коли користувач буде натискати на кнопки редагувати чи видалити (за потреби), то вони спрацьовували тільки на тому полі, яке зв'язане з цими кнопками.

Натискаючи на кнопку редагування відкривається модальне вікно з додаванням інформації про родича, в неї підтягується інформація, яка була введена раніше і збережена в особистому полі відображення. Після виконання редагування при натисканні кнопки «Зберегти інформацію», нова інформація зберігається та записується в своє поле відображення.

При натисканні на кнопку видалення будь-якого поля відображення, це поле видаляється разом з інформацією в ній, всі спеціальні ідентифікатори на інших полях відображення перераховуються та перезаписуються, це реалізовано для правильного подальшого запису інформації в базу даних.

```
<?php
// Витягаємо із бази перелік архівів
$query = "SELECT id_archive,full_name_archive FROM `archives` LIMIT 0,51";
$result = $pdo->query($query); //Виконання запиту
$rows = $result->fetchAll(PDO::FETCH_ASSOC); //Збереження всіх результатів запиту в масиві
?>
```

Рис. 2.37. Кодова частина списку архівних установ

На рисунку 2.37 зображено кодову частину за допомогою якої виконується формування списку з архівними установами, для простішого вводу даних про репресовану особу користувачем. Цей процес реалізується завдяки PHP-запиту до бази даних та перебору даних в ній стосовно цих архівних установ.

```

<?php
foreach($rows as $row_num => $row_value) {
    if(!$row_num) {
        echo '<option selected disabled value="">не обрано</option>';
    }
    echo '<option value="' . $row_value['id_archive'] . '">' . $row_value['full_name_archive'] . '</option>';
}
?>

```

Рис. 2.38 Кодова частина відображення списку архівних установ

На рисунку 2.38 зображена кодова частина, завдяки якій відбувається відображення випадуючого списку зі всіма доступними архівними установами. Інформація виводиться у вигляді масиву із комірок «full_name_archive».

Обробник, який відповідає за запис інформації до бази даних показано на рисунку 2.39.

```

<?php
ini_set('error_reporting', E_ALL);
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);

//Підключаємося до бази даних
require_once("../data.php");
require_once("../db_connect.php");

$lastname = trim($_POST['lastname']);
$firstname = trim($_POST['firstname']);
$patronymic = trim($_POST['patronymic']);

$gender = trim($_POST['repressed_gender']);
$nac_1 = trim($_POST['nationality-1']);
if(!empty($_POST['nationality-2'])) { $nac_2 = $_POST['nationality-2']; } else { $nac_2 = NULL; }

if(!empty($_POST['date_birth_d'])) { $date_birth_d = $_POST['date_birth_d']; } else { $date_birth_d = NULL; }
if(!empty($_POST['date_birth_m'])) { $date_birth_m = $_POST['date_birth_m']; } else { $date_birth_m = NULL; }
if(!empty($_POST['date_birth_y'])) { $date_birth_y = trim($_POST['date_birth_y']); } else { $date_birth_y = NULL; }

$place_of_birth = trim($_POST['place_of_birth']);
$education_info = trim($_POST['education_info']);
$occupation = trim($_POST['occupation']);
$political_views = trim($_POST['political_views']);

if(!empty($_POST['date_death_d'])) { $date_death_d = $_POST['date_death_d']; } else { $date_death_d = NULL; }
if(!empty($_POST['date_death_m'])) { $date_death_m = $_POST['date_death_m']; } else { $date_death_m = NULL; }
if(!empty($_POST['date_death_y'])) { $date_death_y = trim($_POST['date_death_y']); } else { $date_death_y = NULL; }
$place_of_death = trim($_POST['place_of_death']);

if(!empty($_POST['date_detention_d'])) { $date_detention_d = $_POST['date_detention_d']; } else { $date_detention_d = NULL; }
if(!empty($_POST['date_detention_m'])) { $date_detention_m = $_POST['date_detention_m']; } else { $date_detention_m = NULL; }
if(!empty($_POST['date_detention_y'])) { $date_detention_y = trim($_POST['date_detention_y']); } else { $date_detention_y = NULL; }
$location_detention = trim($_POST['location_detention']);

$article_1 = trim($_POST['article_1']);
if(!empty($_POST['article_2'])) { $article_2 = $_POST['article_2']; } else { $article_2 = NULL; }
if(!empty($_POST['article_3'])) { $article_3 = $_POST['article_3']; } else { $article_3 = NULL; }
if(!empty($_POST['article_4'])) { $article_4 = $_POST['article_4']; } else { $article_4 = NULL; }
if(!empty($_POST['article_5'])) { $article_5 = $_POST['article_5']; } else { $article_5 = NULL; }

if(!empty($_POST['date_sentence_d'])) { $date_sentence_d = $_POST['date_sentence_d']; } else { $date_sentence_d = NULL; }
if(!empty($_POST['date_sentence_m'])) { $date_sentence_m = $_POST['date_sentence_m']; } else { $date_sentence_m = NULL; }
if(!empty($_POST['date_sentence_y'])) { $date_sentence_y = trim($_POST['date_sentence_y']); } else { $date_sentence_y = NULL; }
if(!empty($_POST['sentence_org'])) { $sentence_org = $_POST['sentence_org']; } else { $sentence_org = NULL; }
$sentence = trim($_POST['sentence']);

```

Рис. 2.39. Кодова частина обробника для запису(ч.1)

Підключення до бази даних відбувається в на початку файлу (показано на рис. 2.39.). Після чого відбувається перевірка всіх змінних, які використовуються у веб-формі для наповнення бази даних.

```

$add = $pdo->prepare("INSERT INTO test_um_cards(id_archive, id_fond, id_inventory, id_case, repressed_lastname, repressed_firstname, repressed_patronymic, repressed_gender, id_nationality, id_nationality_alternative, id_biograph_info, location_detention, date_detention_y, date_detention_m, date_detention_d, id_authority, date_sentence_y, date_sentence_m, date_sentence_d, sentence, location_residence, rehab_info, id_user, visible) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
$add->execute([$id_archive, $fond, $inventory, $case, $lastname, $firstname, $patronymic, $gender, $nac_1, $nac_2, null, $location_detention, $date_detention_y, $date_detention_m, $date_detention_d, $sentence_org, $date_sentence_y, $date_sentence_m, $date_sentence_d, $sentence, $location_residence, $rehab_info, 99, 1]);
$id_card = $pdo->lastInsertId();

if(!$add) {
    echo "Помилка додавання інформації в таблицю "test_um_cards"!";
} else {
    echo "Картку успішно додано! [test_um_cards]<br /><b>id_card = </b> . $id_card . <br />";
    $result = $pdo->query("SELECT * FROM `test_um_cards` WHERE `id_card` = $id_card");
    $result = $result->fetchAll();
    echo <pre>;
    print_r($result[0]);
    echo </pre>;
}

```

Рис. 2.40. Кодова частина обробника для запису(ч.2)

На рисунку 2.40 показано кодову частину обробника, яка відповідає за запис введених даних у веб-формі в таблицю бази даних «um_cards». Тут перевіряється кількість змінних та їх співпадіння, отримується значення для конкретних комірок в таблиці ті відбувається запис нової карти в базі даних.

```

$check_biogr_data = check_variables(['place_of_birth', 'date_birth_y', 'date_birth_m', 'date_birth_d', 'education_info', 'political_views', 'occupation', 'place_of_death', 'date_death_y', 'date_death_m', 'date_death_d']);
if($check_biogr_data) {
    $add = $pdo->prepare("INSERT INTO test_um_biographical_data(id_card, place_of_birth, date_birth_y, date_birth_m, date_birth_d, education_info, political_views, occupation, place_of_death, date_death_y, date_death_m, date_death_d) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
    $add->execute([$id_card, $place_of_birth, $date_birth_y, $date_birth_m, $date_birth_d, $education_info, $political_views, $occupation, $place_of_death, $date_death_y, $date_death_m, $date_death_d]);
    $id_biographical_data = $pdo->lastInsertId();

    $pdo->query("UPDATE `test_um_cards` SET `id_biograph_info` = '$id_biographical_data' WHERE `test_um_cards`.`id_card` = $id_card");

    if(!$add) {
        echo "Помилка додавання інформації в таблицю "test_um_biographical_data"!";
    } else {
        echo "Також створено запис з "Еібіографічними даними"! [test_um_biographical_data]<br /><b>id_biographical_data = </b> . $id_biographical_data . <br />";
        $result = $pdo->query("SELECT * FROM `test_um_biographical_data` WHERE `id_biographical_data` = $id_biographical_data");
        $result = $result->fetchAll();
        echo <pre>;
        print_r($result[0]);
        echo </pre>;

        echo <b>id_biographical_data</b> додано до картки..";
        $result = $pdo->query("SELECT * FROM `test_um_cards` WHERE `id_card` = $id_card");
        $result = $result->fetchAll();
        echo <pre>;
        print_r($result[0]);
        echo </pre>;
    }
}

```

Рис. 2.41. Кодова частина обробника для запису(ч.3)

На рисунку 2.41 показано кодову частину обробника, яка відповідає за запис введення даних у веб-формі в таблицю бази даних «um_biographical_data». Тут перевіряється кількість змінних та їх співпадіння, отримується значення для конкретних комірок в таблиці ті відбувається запис нової карти в базі даних.

3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

3.1 Результати модернізації бази даних

Створення модернізованої бази даних виявилось непростю задачею. В ході модернізації було побудована нова база даних.

um_articles	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_authorities	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_biographical_data	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_cards	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_kinsman	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_links_cards_articles	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_messages	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_nationalities	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить
um_property	★	Обзор	Структура	Поиск	Вставить	Очистить	Удалить

Рис. 3.1. Структура модернізованої бази даних

Після аналізу старої було вирішено змінити структуру бази даних. На рисунку 3.1. зображена нова структура бази даних. Як бачимо назви таблиць бази даних було змінено на більш зрозумілі та більш точно описуючі, про що саме в них зберігається інформація.

Структури нових модернізованих таблиць показані на рис. 3.2., рис.3.3., рис. 3.4., рис. 3.5., рис. 3.6., рис. 3.7., рис. 3.8., рис. 3.9 та рис. 3.10.

1	id_article	int(11)	Нет	Нет	Унікальний цифровий код (ідентифікатор) статті
2	article_label	text utf8mb4_unicode_ci	Нет	Нет	Позначення статті. Приклад: 54-1а КК УРСР
3	article_text	text utf8mb4_unicode_ci	Нет	Нет	Зміст статті. Зазначається у разі наявності та відповідно до чинного на час винесення вироку Карного кодексу
4	visible	int(11)	Нет	Нет	Статус відображення: 1 – відобразити, 0 – приховати

Рис. 3.2. Структура таблиці «um_articles»

1	<u>id_authority</u> 	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) органу, який виносив вироки
2	<u>shortName_authority</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Повне найменування органу, який виносив вироки
3	<u>fullName_authority</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Скорочена назва органу, який виносив вироки. У разі офіційного застосування
4	<u>visible</u>	int(2)	Нет	Нем	Статус відображення: 1 – відобразити, 0 – приховати

Рис. 3.3. Структура таблиці «um_authorities»


1	<u>id_biographical_data</u> 	int(20)	Нет	Нем	
2	<u>id_card</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) картки репресованого
3	<u>place_of_birth</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Місце народження
4	<u>date_birth_y</u>	int(20)	Да	NULL	Рік народження
5	<u>date_birth_m</u>	int(20)	Да	NULL	Місяць народження
6	<u>date_birth_d</u>	int(20)	Да	NULL	День народження
7	<u>education_info</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Інформація про освіту
8	<u>political_views</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Політичні погляди
9	<u>occupation</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Рід заняття
10	<u>place_of_death</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Місце смерті
11	<u>date_death_y</u>	int(20)	Да	NULL	Рік смерті
12	<u>date_death_m</u>	int(20)	Да	NULL	Місяць смерті
13	<u>date_death_d</u>	int(20)	Да	NULL	День смерті

Рис. 3.4. Структура таблиці «um_biographical_data»

1	<u>id_card</u> 🔑	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) картки репресованого
2	<u>id_archive</u>	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) архівної установи
3	<u>id_fond</u>	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) фонду
4	<u>id_inventory</u>	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) опису
5	<u>id_case</u>	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) справи
6	<u>repressed_lastname</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Прізвище репресованого
7	<u>repressed_firstname</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Ім'я репресованого
8	<u>repressed_patronymic</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	По-батькові репресованого
9	<u>repressed_gender</u>	int(20)	Нет	Нem	Стать репресованого: 0 – невідомо, 1 - чоловіча, 2 – жіноча
10	<u>id_nationality</u>	int(20)	Нет	Нem	Унікальний цифровий код (ідентифікатор) національності
11	<u>id_nationality_alternative</u>	int(20)	Да	NULL	Унікальний цифровий код (ідентифікатор) альтернативної національності
12	<u>id_biograph_info</u>	int(20)	Да	NULL	Унікальний цифровий код (ідентифікатор) біографічних даних
13	<u>location_detention</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Місце проживання на момент арешту

Рис. 3.5. Структура таблиці «um_cards»(ч.1)

14	<u>date_detention_y</u>	int(20)	Да	NULL	Рік арешту
15	<u>date_detention_m</u>	int(20)	Да	NULL	День арешту
16	<u>date_detention_d</u>	int(20)	Да	NULL	Місяць арешту
17	<u>id_authority</u>	int(20)	Да	NULL	Унікальний цифровий код (ідентифікатор) органу винесення вироку
18	<u>date_sentence_y</u>	int(20)	Да	NULL	Рік вироку
19	<u>date_sentence_m</u>	int(20)	Да	NULL	Місяць вироку
20	<u>date_sentence_d</u>	int(20)	Да	NULL	День вироку
21	<u>sentence</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Вирок
22	<u>location_residence</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Місце реабілітації
23	<u>rehab_info</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нem	Інформація про реабілітацію
24	<u>id_user</u>	int(20)	Нет	Нem	Ідентифікатор користувача, який створив картку
25	<u>visible</u>	int(20)	Нет	Нem	Статус відображення: 1 – відобразити, 0 – приховати

Рис. 3.6. Структура таблиці «um_cards»(ч.2)

1	<u>id_kinsman</u>	int(11)	Нет	Нем	
2	<u>id_card</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) картки репресованого
3	<u>kinsman_lastname</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Прізвище родича репресованого
4	<u>kinsman_firstname</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Ім'я родича репресованого
5	<u>kinsman_patronymic</u>	varchar(254) utf8mb4_unicode_ci	Да	NULL	По-батькові родича репресованого
6	<u>kinsman_birth_year</u>	varchar(254) utf8mb4_unicode_ci	Да	NULL	Рік народження родича
7	<u>kinsman_belonging</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Рідство. Приклад: син, брат і т.д.
8	<u>id_card_kinsman</u>	int(20)	Да	NULL	Унікальний цифровий код (ідентифікатор) картки репресованого родича. Зазначається, якщо родича самого було репресовано
9	<u>display_order</u>	int(2)	Нет	Нем	Порядок відображення ролей. Число, починаючи від нуля

Рис. 3.7. Структура таблиці «um_kinsman»

1	<u>id_link</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) зв'язку карток зі статтями вироку
2	<u>id_card</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) картки репресованого
3	<u>id_article</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) статті
4	<u>display_order</u>	int(2)	Нет	Нем	Порядок відображення ролей. Число, починаючи від нуля

Рис. 3.8. Структура таблиці «um_links_cards_articles»

1	<u>id_nationality</u>	int(20)	Нет	Нем	Унікальний цифровий код (ідентифікатор) національності
2	<u>male_value</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Написання національності чоловіків
3	<u>female_value</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Написання національності жінок
4	<u>description</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нем	Описання. Зазначається за потребою
5	<u>visible</u>	int(2)	Нет	Нем	Статус відображення: 1 – відобразити, 0 – приховати

Рис. 3.9. Структура таблиці «um_nationalities»

1	<u>id_property</u>	int(11)	Нет	Нет	Унікальний цифровий код (ідентифікатор) код одиниці майна
2	<u>id_card</u>	int(20)	Нет	Нет	Унікальний цифровий код (ідентифікатор) картки репресованого
3	<u>property_description</u>	varchar(254) utf8mb4_unicode_ci	Нет	Нет	Описування майна
4	<u>display_order</u>	int(2)	Нет	Нет	Порядок відображення ролей. Число, починаючи від нуля

Рис. 3.10. Структура таблиці «um_property»

3.2 Результати роботи створеної веб-форми

Усього було виконано дві задачі: створення веб-форми для вводу даних про репресованих осіб, із зручним та інтуїтивно зрозумілим інтерфейсом; написання обробника веб-форми для коректного відображення у браузері та коректного запису інформації до бази даних.

Веб-форма для введення даних показана на рис. 3.11., рис. 3.12., рис. 3.13.

Український мартиролог ХХ ст.

[додавання нової картки]

Загальна інформація

<input type="text" value="Прізвище"/>	<input type="text" value="Ім'я"/>	<input type="text" value="По-батькові"/>	<input type="text" value="Стать: не обрано"/>
<input type="text" value="Національність: не обрано"/>	<input type="button" value="+"/>		

Бібліографічні дані

<input type="text" value="Дата народження: День"/>	<input type="text" value="Місяць"/>	<input type="text" value="Рік"/>	<input type="text" value="Місце народження"/>
<input type="text" value="Освіта"/>			
<input type="text" value="Рід занять"/>			
<input type="text" value="Політичні погляди"/>			
<input type="text" value="Дата смерті: День"/>	<input type="text" value="Місяць"/>	<input type="text" value="Рік"/>	<input type="text" value="Місце смерті"/>

Рис. 3.11. Відображення веб-форми(ч.1)

Інформація про вирок

Дата арешту: День ▼ Місяць ▼ Рік Місце проживання на момент арешту Введіть місце проживання

Стаття #1: не обрано +

Дата вироку: День ▼ Місяць ▼ Рік Орган винесення вироку: не обрано

Вирок: Введіть інформацію про вирок

Інформація про реабілітацію

Місце реабілітації Введіть інформацію про місце реабілітації

Відомості про реабілітацію Введіть відомості про реабілітацію

Інформація про майно

Одиниця майна #1 Введіть відомості про одиницю майна +

Рис. 3.12. Відображення веб-форми(ч.2)

Інформація про родичів

Пошук в базі даних: Пошук родича в базі даних

Додати інформацію про родича вручну

Архівні відомості

Архівна установа: не обрано

№ фонду Введіть номер фонду № опису Введіть номер опису № справи Введіть номер справи

+ Додати картку до БД

Рис. 3.13. Відображення веб-форми(ч.3)

У веб-формі форми було реалізовано валідацію. Це потрібно для того, щоб під час заповнення форми, користувачу потрібно було заповнити певні поля для вводу даних, тим самим буде уникнуто варіанту створення пустих карток в базі. Без такої функції база даних буде некоректно працювати, її розміри будуть значно більше, за рахунок неправильно введених даних.

На рис. 3.14., рис. 3.15., рис. 3.16 показано веб-форму після спроби збереження інформації в базу даних при некоректному вводі інформації до обов'язкових полів вводу.

Український мартиролог ХХ ст. [додавання нової картки]

Загальна інформація

Прізвище: ⓘ
Будь-ласка, введіть коректне прізвище.

Ім'я: ⓘ
Будь-ласка, введіть коректне ім'я.

По-батькові: ✓

Стать: ⓘ ▼
Будь-ласка, оберіть стать, або поставте "_невідомо".

Національність: ⓘ ▼ +
Будь-ласка, оберіть національність, або поставте "_невідомо".

Бібліографічні дані

Дата народження: ✓ ▼ ✓ ▼ ✓ ▼ ✓

Освіта: ✓

Рід занять: ✓

Політичні погляди: ✓

Дата смерті: ✓ ▼ ✓ ▼ ✓ ▼ ✓

Рис. 3.14. Відображення веб-форми з валідацією(ч.1)

Інформація про вирок

Дата арешту: ✓ ▼ ✓ ▼ ✓ ▼ ✓

Стаття #1: ⓘ ▼ +
Будь-ласка, оберіть статтю, або поставте "_невідомо".

Стаття #2: ✓ ▼ ×

Дата вироку: ✓ ▼ ✓ ▼ ✓ ▼ ⓘ ▼
Будь-ласка, оберіть орган винесення вироку, або поставте "_невідомо".

Вирок: ✓

Інформація про реабілітацію

Місце реабілітації: ✓

Відомості про реабілітацію: ✓

Інформація про майно

Одиниця майна #1: ✓ +

Рис. 3.15. Відображення веб-форми з валідацією (ч.2)

Інформація про родичів

Пошук в базі даних: ✓ Пошук родича в базі даних

Додати інформацію про родича вручну

Архівні відомості

Архівна установа: ⓘ ▼

Будь-ласка, оберіть архівну установу, або поставте "_невідомо".

№ фонду ⓘ № опису ⓘ № справи ⓘ

Будь-ласка, введіть № фонду. Будь-ласка, введіть № опису. Будь-ласка, введіть № справи.

+ Додати картку до БД

Рис. 3.16. Відображення веб-форми з валідацією (ч.3)

Після введення всіх потрібних даних та перевірку їх коректності, вся інформація записується в базу даних.

Після запису всіх даних у базу даних веб-форма відкриває нове вікно в браузері, в якому показано які дані та куди записуються(назва таблиці, комірки). Ця можливість надана особам, які мають прямий доступ до бази даних, та мають право модерувати роботу з базою даних та веб-формою.

На рис. 3.17., рис. 3.18 показано сторінка з демонстрацією даних для модераторів.

Картку успішно додано! [test_um_cards]
id_card = 1

Array

```
(
  [id_card] => 1
  [id_archive] => 30
  [id_fond] => 152
  [id_inventory] => 23
  [id_case] => 3
  [repressed_lastname] => Гавриленко
  [repressed_firstname] => Пертро
  [repressed_patronymic] => Іфанович
  [repressed_gender] => 1
  [id_nationality] => 1
  [id_nationality_alternative] => 2
  [id_biograph_info] =>
  [location_detention] =>
  [date_detention_y] => 1931
  [date_detention_m] => 3
  [date_detention_d] => 12
  [id_authority] => 1
  [date_sentence_y] => 1932
  [date_sentence_m] => 4
  [date_sentence_d] => 9
  [sentence] =>
  [location_residence] =>
  [rehab_info] =>
  [id_user] => 99
  [visible] => 1
)
```

Також створено запис з "Бібліографічними даними"! [test_um_biographical_data]
id_biographical_data = 1

Array

```
(
  [id_biographical_data] => 1
  [id_card] => 1
  [place_of_birth] => м. Харків
  [date_birth_y] => 1920
  [date_birth_m] => 6
  [date_birth_d] => 3
  [education_info] => середня
  [political_views] =>
  [occupation] => будівельник
  [place_of_death] => Полтава
  [date_death_y] => 1976
  [date_death_m] => 10
  [date_death_d] => 16
)
```

Рис. 3.17. Відображення записаних даних(ч.1)

id_biographical_data додано до картки..

```

Array
(
    [id_card] => 1
    [id_archive] => 30
    [id_fond] => 152
    [id_inventory] => 23
    [id_case] => 3
    [repressed_lastname] => Гавриленко
    [repressed_firstname] => Пертро
    [repressed_patronymic] => Іфанович
    [repressed_gender] => 1
    [id_nationality] => 1
    [id_nationality_alternative] => 2
    [id_biograph_info] => 1
    [location_detention] =>
    [date_detention_y] => 1931
    [date_detention_m] => 3
    [date_detention_d] => 12
    [id_authority] => 1
    [date_sentence_y] => 1932
    [date_sentence_m] => 4
    [date_sentence_d] => 9
    [sentence] =>
    [location_residence] =>
    [rehab_info] =>
    [id_user] => 99
    [visible] => 1
)

```

Рис. 3.18. Відображення записаних даних(ч.2)

Результати запису в базу даних phpMyAdmin показано на рис. 3.19., рис. 3.20., рис. 3.21. та рис. 3.22.

id_biographical_data	id_card Унікальний цифровий код (идентифікатор) картки реп...	place_of_birth Місце народження	date_birth_y Рік народження	date_birth_m Місяць народження	date_birth_d День народження	education_info Інформація про освіту	political_views Політичні погляди	occupation Рід заняття	place_of_death Місце смерті	date_death_y Рік смерті	date_death_m Місяць смерті	date_death_d День смерті
1	1	м. Харків	1920	6	3	середня		будівельник	Полтава	1976	10	16

Рис. 3.19. Запис в базі даних в таблиці «um_biographical_data»

id_card	id_archive	id_fond	id_inventory	id_case	repressed_lastname	repressed_firstname	repressed_patronymic	repressed_gender	id_nationality	id_nationality_alternative	id_biograph_info	location_detention
Унікальний цифровий код (идентифікатор) картки реп...	Унікальний цифровий код (идентифікатор) архівної у...	Унікальний цифровий код (идентифікатор) фонду	Унікальний цифровий код (идентифікатор) опису	Унікальний цифровий код (идентифікатор) справи	Прізвище репресованого	Ім'я репресованого	По-батькові репресованого	Стать репресованого: 0 – невідомо, 1 - чоловік, 2...	Унікальний цифровий код (идентифікатор) національ...	Унікальний цифровий код (идентифікатор) альтернати...	Унікальний цифровий код (идентифікатор) біографіч...	Місце проживання на момент арешту
1	30	152	23	3	Гавриленко	Пертро	Іфанович	1	1	2	1	

Рис. 3.20. Запис в базі даних в таблиці «um_cards»(ч.1)

location_detention	date_detention_y	date_detention_m	date_detention_d	id_authority	date_sentence_y	date_sentence_m	date_sentence_d	sentence	location_residence	rehab_info	id_user	visible
Місце провадження на момент арешту	Рік арешту	День арешту	Місяць арешту	Унікальний цифровий код (ідентифікатор) органу вик...	Рік вироку	Місяць вироку	День вироку	Вирок	Місце реабілітації	Інформація про реабілітацію	Ідентифікатор користувача, який створив картку	Статус відображення: 1 – відобразити, 0 – приховати...
	1931	3	12	1	1932	4	9				99	1

Рис. 3.21. Запис в базі даних в таблиці «um_cards»(ч.2)

id_property	id_card	property_description	display_order
Унікальний цифровий код (ідентифікатор) код одиниц...	Унікальний цифровий код (ідентифікатор) картки реп...	Описування майна	Порядок відображення ролей. Число, починаючи від н...
1	1	дім на хуторі	1

Рис. 3.22. Запис в базі даних в таблиці «um_property»

3.3 Рекомендації до отриманих результатів

Проведені експерименти показали можливості веб-форми та взаємодію її з модернізованою базою даних.

Представлена у роботі база даних має просту структуру, що дозволяє їй коректно взаємодіяти з веб-формою для вводу даних. Це дозволяє значно збільшити швидкість наповнення бази даних, завдяки розділенню обов'язків між працівниками.

Як і звичайна форма для вводу, веб-форма для наповнення бази даних «Український мартиролог ХХ століття» має просту структуру із різними полями вводу та іншими елементами для наповнення, але при цьому ці елементами є спеціальними та більш корисними для поставленої задачі.

ВИСНОВКИ

У роботі розкрито питання модернізації бази даних та створення веб-форми для наповнення цієї бази даних. Проведено аналіз застарілої бази даних. Отримано потреби та властивості для модернізації та створення нової бази даних.

У ході виконання даної роботи було досягнуті наступні результати:

- досліджено питання баз даних;
- розкрито питання важливості правильної побудови баз даних;
- отримано вимоги до структури бази даних;
- досліджено принцип роботи веб-форм;
- отримано вимоги до веб-форми;
- створено модернізовану базу даних;
- створено веб-форму для наповнення бази даних;
- з'ясовано позитивний вплив на технічний процес наповнення з використанням веб-форми.

Завдяки отриманим результатам можна підсумувати, що використання модернізованої бази даних та веб-форми для її наповнення, позитивно впливають на процес виконання поставленої задачі перед архівними установами, а саме збереження історичних відомостей.

Отже, використання даного підходу є практично значимим елементом у наповненнях баз даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Что такое PHP? URL: <https://www.php.net/manual/ru/intro-what-is.php> (дата звернення: 09.11.2021).
2. Что такое PHP и зачем это нужно? URL: <https://webkurs.info/page/chto-takoe-php-i-zachem-eto-nuzhno> (дата звернення: 10.11.2021).
3. PHP Tutorial URL: <https://www.w3schools.com/php/default.asp> (дата звернення: 10.11.2021).
4. JavaScript URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 10.11.2021).
5. JavaScript Algorithms and Data Structures URL: <https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/#basic-javascript> (дата звернення: 10.11.2021).
6. The Modern JavaScript Tutorial URL: <https://javascript.info/> (дата звернення: 10.11.2021).
7. HTML URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (дата звернення: 10.11.2021).
8. Что такое HTML? Основы языка разметки гипертекста URL: <https://www.hostinger.com.ua/rukovodstva/shto-takoe-html/> (дата звернення: 10.11.2021).
9. HTML Tutorials URL: <https://www.geeksforgeeks.org/html-tutorials/> (дата звернення: 10.11.2021).
10. II МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ
«TELECOMMUNICATION: PROBLEMS AND INNOVATION. Pages 11 -
12. URL: http://www.dut.edu.ua/uploads/1_2187_52947573.pdf (дата звернення: 10.11.2021).
11. HTML computer science URL: <https://www.britannica.com/technology/HTML> (дата звернення: 10.11.2021).

12. Что такое CSS: объясняем простыми словами URL:

<https://gb.ru/posts/что-такое-css-объясняем-простыми-словами> (дата
звернения: 11.11.2021).

13. MySQL URL: <https://www.mysql.com/> (дата звернения: 11.11.2021).

14. Руководство по MySQL URL: <https://metanit.com/sql/mysql/> (дата
звернения: 11.11.2021).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ