МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н. Каразіна
Факультет радіофізики, біомедичної електроніки та комп'ютерних систем

«ЗАТВЕРДЖУЮ»
завідувач кафедри фізичної
і біомедичної електроніки та
комплексних інформаційних
технологій
доктор фіз.-мат. наук, доцент
_____ Сергій БЕРДНИК
«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
МАГІСТРА

на тему:

МОДУЛЬ МЕРЕЖЕВОГО ЗБЕРІГАННЯ ПУЛЬМОНОЛОГІЧНИХ ДАНИХ

Виконав: студент ІІ курсу магістратури, групи РЕ-61
спеціальності 153 «Мікро- та наносистемна техніка»
освітньо-професійна програма «Фізична та біомедична електроніка»

_____ Олег КРИВОГУЗ

Науковий керівник:
д-р фіз.-мат. наук, с.н.с., професор _____ Віктор КАТРИЧ

Харків – 2023

РЕФЕРАТ

Кривогуз О.О. Модуль мережевого зберігання пульмонологічних даних. Дипломна робота. Харківський національний університет імені В. Н. Каразіна, 2023. 79 с., 21 рис., 9 табл., 7 дод., 25 джерел.

КАРТКА ПАЦІЄНТА ЕЛЕКТРОННА, СИСТЕМА ДИХАННЯ, ЕМНІСТЬ ДИХАЛЬНА, ОБ'ЄМ ДИХАЛЬНИЙ, ЗОБРАЖЕННЯ РЕНТГЕНІВСЬКЕ, БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗОЮ ДАНИХ, SQL-ЗАПИТ.

Об'єкт дослідження: електронна картка пацієнта.

Предмет дослідження: особливості використання програмного забезпечення для модуля мережевого зберігання пульмонологічих даних.

Мета роботи: розробка програмного забезпечення для модуля мережевого зберігання пульмонологічих даних.

Методи дослідження: спірометрія, рентгенографія.

Результати дослідження: Розглянуто функціонально-анатомічні особливості системи дихання, інструментальні методи їх дослідження. Обрано набір показників для зберігання за допомогою модулю. Розроблена база даних, запропонована структура онлайн-системи управління базою даних та схема взаємодії програмних блоків, на базі якої було виконана програмна реалізація з використанням *php 5.3, MySQL 5.1, HTML 5.0*.

Розроблений модуль може бути використаний як складова у складі мережевої інформаційної клінічної системи.

ABSTRACT

Kryvoguz O.O. Module for network storage of pulmonology data. Graduate work. V. N. Karazin Kharkiv National University, 2023. 79 pages, 21 figures, 9 tables, 7 appendices, 25 references.

ELECTRONIC PATIENT CARD, RESPIRATORY SYSTEM, RESPIRATORY CAPACITY, RESPIRATORY VOLUME, X-RAY IMAGE, DATABASE, DATABASE MANAGEMENT SYSTEM, SQL QUERY.

Research object: electronic card of the patient.

The subject of the study: peculiarities of using the software for the network storage module of pulmonology data.

The aim of the work: development of software for the network storage module of pulmonology data.

Research methods: spirometry, radiography.

Research results: Functional and anatomical features of the respiratory system, instrumental methods of their research are considered. A set of indicators is selected for storage using the module. The database was developed, the structure of the online database management system and the interaction scheme of the software blocks were proposed, on the basis of which the software implementation was performed using php 5.3, MySQL 5.1, HTML 5.0.

The developed module can be used as a component of the network information clinical system.

ЗМІСТ

ВСТУП……		6
РОЗДІЛ 1. МЕДИКО-ТЕХНІЧНЕ ОБГРУНТУВАННЯ……		7
	1.1. Фізіологія видиху……	7
	1.2. Легеневі об’єми та ємності…….	18
	1.3. Основні методи досліджень у пульмонології…….....	25
	1.3.1. Спірометрія…….....	25
	1.3.2. Рентгенографія…….	27
	1.3.3. Томографічні методи…….....	27
	1.4. Аналіз ринку сучасних медичних інформаційних систем……..	29
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ……		34
	2.1. Структура бази даних…….	34
	2.2. Взаємодія модулів СУБД……...	38
	2.3. Інтерфейс СУБД……..	39
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ……		41
	3.1. Програмні засоби……	41
	3.2. Створення бази даних…….	42
	3.3. Головна сторінка…….	42
	3.4. Меню……	43
	3.5. Пошук історії хвороби……	44
	3.5.1. Пошук в таблиці **«patient»**……..	44
	3.5.2. Пошук в таблиці **«visit»**……...	52
	3.5.3. Пошук в таблицях **«spiro»** і **«xray»**……	55
	3.6. Створення нової карти пацієнта……	60
	3.7. Видалення карти пацієнта……..	65
РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ……		67
	4.1. Опис програмного продукту……..	67
	4.2. Оцінка ринку збуту…….	68

	4.3. Розрахунок затрат на розробку ПП……...	68
	4.4. Оцінка рівня конкурентоспроможності ПП…….	74
	4.5. Оцінка ризику ПП……...	76
ВИСНОВКИ…….....		79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……..		80
Додаток А……...		83
Додаток Б……		84
Додаток В……		85
Додаток Г……		86
Додаток Д……		87
Додаток Е……		88
Додаток Ж……...		89

ВСТУП

Сучасний стан ринку медичних інформаційних систем такий, що більше 60% існуючих зразків відноситься до типу комп'ютерних діагностичних, 10% складають системи для виконання адміністративних завдань, 20% - системи автоматизованих робочих місць лікаря, 10% - системи вузькопрофільного спеціалізованого призначення [1, 2, 7, 8].

Понад 50 % їх – морально застарілі системи, розроблені понад 20 років і відповідають сучасним вимогам. В умовах розвитку медичних технологій, і особливо в галузях функціональної діагностики, як пульмонологія, а також внаслідок прогресу комп'ютерних способів обробки та зберігання даних, актуальним є завдання створення медичних інформаційних систем підтримки прийняття рішення лікарем [9, 13, 16, 17]. Назріла необхідність як механічного ведення історій хвороб, а й аналізу даних, отриманих під час лікування. Крім того, сучасні технології дозволяють зберігати дані не тільки у текстовому вигляді, а й у вигляді зображень, що буває особливо цінним при стеженні за розвитком захворювання та ефективністю терапії.

У основній частині дипломної роботи розглядається можливість створення медичної інформаційної системи пульмонологічних хворих. Для виконання поставленої мети необхідно вирішити низку завдань:

- вивчити фізіологію дихання, основні показники зовнішнього дихання, розглянути інструментальні методи їх дослідження;
- проаналізувати ринок медичних інформаційних систем та вимог, що пред'являються до них;
- синтезувати схему модуля та визначити структуру її елементів;
- розробити базу даних та інтерфейс системи управління базою даних;
- розробити алгоритмічне та програмне забезпечення модуля.

Завданням економічної частини є техніко-економічне обґрунтування програмного продукту, розробленого у межах атестаційної роботи.

РОЗДІЛ 1. МЕДИКО-ТЕХНІЧНЕ ОБГРУНТУВАННЯ

1.1. Фізіологія видиху.

Видих – це один із основних процесів дихання, який відбувається у легенях організму. Під час видиху, легені видаляють використане повітря з організму, а разом із ним видаляються вуглекислий газ та інші продукти обміну речовин [3, 4, 5].

Дихальна система складається з ряду органів і структур, які працюють спільно для забезпечення процесу дихання та обміну газів [15, 18, 22].

Основні складові дихальної системи включають:

1. Носова порожнина (носові проходи): основна функція носової порожнини - очищення, зігрівання та зволоження повітря, що вдихається перед тим, як воно потрапить до легенів.
2. Гортань: з'єднує носову порожнину і трахею. Гортань відіграє роль у регуляції звуку та уникнення проникнення їжі в дихальні шляхи.
3. Трахея: трубка, яка переносить повітря від гортані до бронхів та далі до легенів.
4. Бронхи та їх розділення (бронхіоли): трахея розгалужується на два бронхи, які поділяються на дедалі менші бронхіоли. Ці структури приводять повітря до альвеол, де відбувається газообмін.
5. Легені: органи, що містять альвеоли (мішки для газообміну), де кисень переходить з повітря в кров, а вуглекислий газ переходить з крові в повітря. Легені забезпечують основний процес газообміну між організмом і зовнішнім середовищем.
6. Діафрагма та м’язи міжреберні: м'язи, що відповідають за випадіння та розширення грудної клітини. Діафрагма розділяє груди і черево, контролюючи об'єм легенів.

Процес видиху розпочинається після того, як м’язи діафрагми та міжреберні м'язи скорочуються, зменшуючи об'єм грудей. Це призводить до

збільшення тиску у легенях порівняно з атмосферним тиском, і повітря починає виходити з легень через дихальні шляхи.

Окрім цього основного механізму, видих включає активність різних дихальних м'язів та регуляцію діаметру дихальних шляхів. У процесі видиху також відбувається важливий обмін газів. Вуглекислий газ, який утворюється в результаті обміну речовин, переходить з крові в повітря легень і потім видаляється з організму.

Газообмін в альвеолах є ключовим процесом дихання, де відбувається обмін кисню (O2) та вуглекислого газу (CO2) між повітрям у легенях і кров'ю в капілярах. Цей процес відбувається в результаті дифузії – переміщення молекул газів від області вищого тиску до області нижчого тиску (рис. 1.1).

Загалом, видих – це важливий етап дихального процесу, який дозволяє організму виводити продукти обміну газів та забезпечити нормальну газообмінну функцію в легенях.

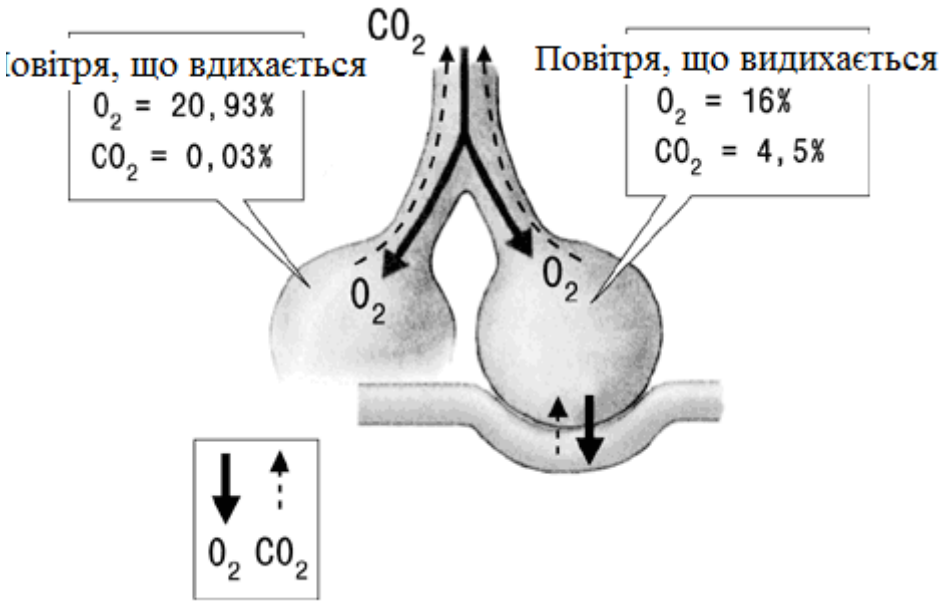

Рис. 1.1. Газообмін в альвеолах

Під час дихання, процес дифузії є ключовим механізмом для обміну газів між атмосферним повітрям і кров'ю в легенях. Розглянемо основні етапи дифузії:

Вдихання: під час вдиху повітря, що містить кисень, потрапляє в носову порожнину, далі через гортань, трахею і бронхи до альвеол легенів.

Дифузія кисню в альвеолах: у альвеолах, де відбувається газообмін, концентрація кисню в повітрі в альвеолах вища, ніж в крові. Таким чином, кисень потрапляє з повітря альвеол у кров через тонку стінку альвеол та стінку капілярів.

Приєднання кисню до гемоглобіну: коли кисень потрапляє у кров, він приєднується до гемоглобіну, який знаходиться в еритроцитах. Це утворює оксигемоглобін – хімічну сполуку, що легко переноситься кров'ю.

Транспорт кисню в тіла: оксигемоглобін переноситься еритроцитами усюди по організму за допомогою кров'яного потоку. На рівні тканин відбувається зворотний процес – дифузія кисню з гемоглобіну в клітини.

Дифузія вуглекислого газу: вуглекислий газ, який утворюється в клітинах у результаті метаболічних процесів, проникає з клітин у кров'яні капіляри.

Дифузія вуглекислого газу в альвеолах: в альвеолах концентрація вуглекислого газу вища, ніж у повітрі. Таким чином, вуглекислий газ проникає з крові в альвеоли.

Видих: під час видиху вуглекислий газ, який накопичується в альвеолах, виводиться на поверхню через дихальні шляхи.

Стан, при якому клітини організму не отримують достатньої кількості кисню називається – гіпоксія. Це зумовлено низькою концентрацією кисню в повітрі, обмеженим доступом кисню до легень або проблемами з ефективністю транспорту кисню кров'ю до тканин.

Недостатнє надходження кисню до організму може мати різноманітні причини, які включають медичні, оточуючі та інші фактори. Наведемо найбільш поширені причини недостатнього надходження кисню.

Хронічні захворювання легень, такі як хронічний обструктивний захворювання легень, бронхіт, астма або фіброз легенів, можуть обмежити обсяг легеневих альвеол для газообміну. Серцева недостатність або інші серцеві проблеми можуть призводити до недостатнього переносу кисню кров'ю до тканин. Пневмонія або дихальна недостатність можуть порушити газообмін в легенях. Низький рівень гемоглобіну у крові (анемія) може призводити до недостатнього транспорту кисню з легенів до тканин.

Перебування на великих висотах може збільшити рівень атмосферного тиску, що може призвести до менш ефективного надходження кисню в легені. Обмеження дихальних шляхів, таке як утруднене дихання через алергії, опухлість гортані чи інші фактори, може ускладнити процес вдихання кисню. Вдихання токсичних речовин, таких як гази чи дим, може завдати шкоди легеням та обмежити газообмін. Деякі інші стани, такі як недостатність кровопостачання до легенів, аспірація (потрапляння іноземних тіл або рідин в дихальні шляхи), альвеолярний гіповентиляційний синдром та інші, також можуть впливати на газообмін.

В організмі людини існують механізми, за допомогою яких відновлюються або підтримуються фізіологічні функції чи стан. Ці реакції можуть відбуватися на різних рівнях включаючи молекулярний, клітинний, тканинний та ін.

Наприклад, якщо виникає гіпоксія (недостатнє надходження кисню до тканин), організм може активувати компенсаторні механізми для забезпечення якнайбільш ефективного використання доступного кисню.

Наведемо деякі приклади компенсаторних реакцій. Підвищення частоти дихання може допомогти забезпечити більше кисню для легенів та тканин. Розширення артерій допомагає збільшити кількість кисню, який може бути доставлений до тканин. Збільшення серцевого ритму (відповідно - кровопостачання) підтримує ефективний транспорт кисню по організму. Організм може допомогти гемоглобін прийняти форму, яка легше вивільняє

кисень там, де він потрібен. У деяких випадках організм може активувати гени, які сприяють адаптації до зміни умов.

Під час інтенсивного м'язового навантаження збільшується попит організму на кисень, оскільки м'язи активно використовують його для вироблення енергії під час аеробного метаболізму. Процес, в якому м'язи використовують кисень для отримання енергії з глюкози або інших джерел, відомий як аеробний метаболізм.

При фізичній активності відбувається збільшення частоти та глибини дихання для забезпечення більшого потоку кисню в легені. Інтенсивна фізична активність призводить до збільшення серцевого ритму, що забезпечує ефективніше перенесення кисню з легенів до м'язів.

Відбувається розширення артерій для поліпшення кровопостачання до активних м'язів, що допомагає забезпечити достатню кількість кисню. Організм може відповісти на інтенсивне тренування шляхом збільшення об'єму крові та кількості червоних кров'яних тілець, які переносять кисень. Мітохондрії, які відповідають за вироблення енергії у клітинах, можуть бути збільшені та ефективніше функціонувати під впливом тренувань. Зі збільшенням інтенсивності тренування м'язи активують аеробний обмін речовин для отримання енергії з кисню.

Міоглобін - це білок, який має важливе значення для транспорту кисню у м'язових тканинах живого організму. Цей білок є частиною м'язових волокон і відповідає за перенесення та забезпечення киснем, необхідним для енергетичних процесів під час м'язової діяльності.

Головна функція міоглобіну полягає в транспорті кисню в м'язових тканинах. Коли м'язи працюють, вони потребують додаткового кисню для аеробного метаболізму, і міоглобін допомагає переносити цей кисень з легенів до м'язів.

Міоглобін має комплексну структуру та складається з білкового ланцюга. Його основний компонент - гем - містить залізо, що здатне зв'язуватися з молекулою кисню. Міоглобін розташований в м'язових

волокнах і є основним засобом, за допомогою якого м'язи можуть забезпечити себе киснем для енергетичних потреб.

При вище описаних станах організм вмикає регуляцію інтенсивності дихання. Процес, за допомогою якого організм підтримує оптимальний рівень дихальної активності для забезпечення належного обміну газів у легенях.

Цей процес регулюється центральною нервовою системою і залучає різні хімічні та фізіологічні механізми. До основних факторів, які впливають на регуляцію інтенсивності дихання відносяться: рівень вуглекислого газу в крові, рівень кисню в крові, рецептори хімічного чуття, рецептори розтягнення легень, нейромускулярні механізми, психогенні фактори.

Всі ці фактори взаємодіють між собою для підтримки оптимального газообміну в легенях та забезпечення ефективної оксигенації крові. Система регуляції дихання дозволяє організму адаптуватися до змін у середовищі та фізіологічних потребах.

Дихальний центр - це комплексні нейронні структури, відповідальні за регулювання процесів дихання. Ці структури знаходяться у верхній частині спинного мозку та в мозковому стовбурі.

Частота та глибина дихання регулюються дихальним центром у мозковому стовбурі та медулярній області спинного мозку. Ці параметри дихання можуть змінюватися відповідно до потреб організму, рівня фізичної активності, рівнів кисню та вуглекислого газу в крові, а також від інших факторів.

Кількість дихальних циклів, які відбуваються протягом однієї хвилини називається частотою дихання. Вона може змінюватися відповідно до потреб організму. Підвищення частоти дихання забезпечує більше повітря та кисню для організму.

Глибиною дихання називається об'єм повітря, який вдихається та видихається за один дихальний цикл. Глибина дихання може змінюватися

для забезпечення оптимального обміну газів. Підвищення глибини дихання дозволяє вдихнути більше кисню та вивести більше вуглекислого газу.

Регулювання частоти та глибини дихання здійснюється нервовими сигналами, що виходять з дихального центру і йдуть до м'язів дихання, таких як діафрагма та міжреберні м'язи. Ці сигнали контролюють роботу м'язів та визначають, наскільки ефективно відбувається процес дихання. Також важливо враховувати, що інші фактори, такі як емоції, стрес, фізичний стан, патології можуть впливати на ці параметри та спричиняти їхні зміни. Всі ці механізми допомагають забезпечити оптимальний газообмін та використання кисню в організмі.

Регуляція дихання - процес, за допомогою якого організм забезпечує стабільний обмін газів, включаючи кисень і вуглекислий газ, для підтримки нормальної функції клітин та органів. Цей процес включає в себе різні механізми та системи, які реагують на зміни внутрішнього та зовнішнього середовища.

Процес дихання складається з кількох важливих фізіологічних етапів, які взаємодіють для забезпечення обміну газів між організмом та зовнішнім середовищем. Основні складові процесу дихання включають:

Вдихання: дихальні м'язи, зокрема діафрагма та міжреберні м'язи, скорочуються, збільшуючи об'єм грудей. Це призводить до зниження тиску в легенях, і повітря втягується з атмосфери в легені.

Газообмін у легенях: відбувається обмін газів між повітрям та кров'ю. Кисень потрапляє з повітря в альвеоли (мікроскопічні пухлини в легенях) у кров. Вуглекислий газ, утворений в клітинах організму, переходить з крові в повітря для подальшого видихання.

Транспортування кисню кров’ю: кисень, що потрапив в крові в легенях, транспортується через червоні кров'яні клітини (гемоглобін) до клітин та тканин організму.

Видихання: дихальні м'язи розслаблюються, а об'єм грудей зменшується. Це призводить до підвищення тиску в легенях, і повітря

виходить з легенів в атмосферу. Вуглекислий газ, який утворився в клітинах організму, виводиться під час видихання.

Транспортування вуглекислого газу кров’ю: транспортується через кров до легень для виведення його з організму під час видихання.

Ці етапи взаємодіють та забезпечують постійний обмін газів, який необхідний для забезпечення клітин організму достатньою кількістю кисню та виведення зайвого вуглекислого газу.

Дихальний центр представляє сукупність нейронів, розташованих у довгастому мозку центральної нервової системи.

При нормальному диханні центр вдиху посилає ритмічні сигнали до м'язів грудей та діафрагми, стимулюючи їх скорочення. Ритмічні сигнали утворюються внаслідок спонтанного утворення електричних імпульсів нейронами дихального центру.

Скорочення дихальних м'язів призводить до збільшення об'єму грудної порожнини, внаслідок чого повітря входить у легені. У міру збільшення об'єму легень збуджуються рецептори розтягування, які розташовані в стінках легень; вони посилають сигнали у мозок – у центр видиху. Цей центр пригнічує активність центру вдиху і потік імпульсних сигналів до дихальних м'язів припиняється. М'язи розслабляються, об'єм грудної порожнини зменшується, повітря з легких витісняється назовні.

Інтенсивне використання кисню клітинами організму призводить до перевищеного виділення вуглекислого газу, що може призвести до змін у рівнях газів крові. Такий процес може виникнути внаслідок фізичної активності, стресу, захворювань чи інших факторів.

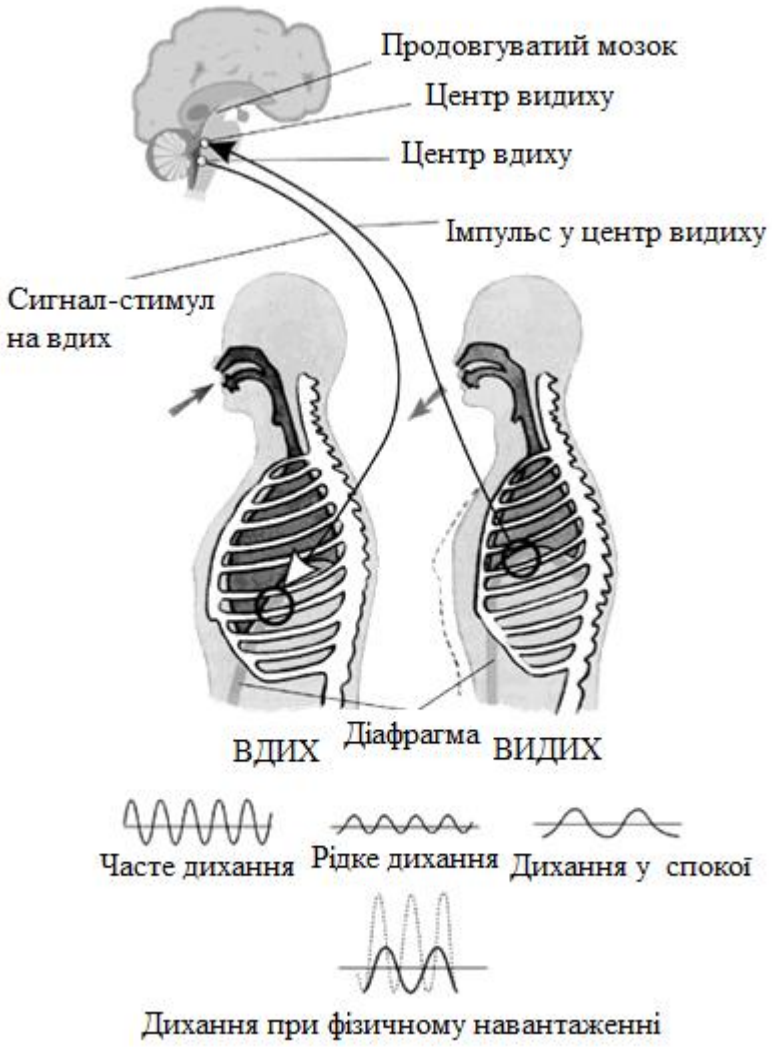

Рис. 1.2. Регуляція дихання

У нормі дихання дихальні цикли представлені рівномірними та координованими процесами вдихання та видихання. Зазвичай дихання відбувається без неприродних перерв чи нерегулярностей. Нормальне дихання базується на наступних характеристиках.

Частота дихання - для дорослих людей, зазвичай, нормальна частота дихання становить близько 12-20 дихальних циклів за хвилину. Це може варіювати в залежності від фізичної активності, емоційного стану та інших факторів.

Глибина дихання – що визначає об'єм повітря, який ви вдихаєте або видихаєте під час одного дихального циклу. Нормальна глибина дихання забезпечує достатній обмін газів у легенях.

Регулярність - нормальне дихання має бути регулярним і координованим. Інтервали між вдиханням і видиханням повинні бути приблизно однаковими.

Безболісність і без напруженість - дихання не повинно супроводжуватися болем або напруженням. Нормальний процес дихання здійснюється без зайвих зусиль.

У стані спокою зазвичай спостерігається нормальний рівень насиченості киснем (оксигену) в крові. Насиченість киснем визначає, який відсоток гемоглобіну у червоних кров'яних тільцях має зв'язаний кисень. Зазвичай цей показник становить близько 95-100%. У цей період, коли потреба організму в кисні не є дуже великою, процес насичення киснем відбувається ефективно. Спокійний стан сприяє рівномірній роботі дихальної та серцево-судинної систем, забезпечуючи оптимальний газообмін в легенях і насиченість киснем у крові.

Інтенсивне дихання забезпечує збільшену постачання кисню крові, що потім транспортується до клітин і тканин організму. Це особливо важливо під час фізичної активності, коли зростає потреба тіла в додатковому кисні для вироблення енергії. Допомагає підтримувати ефективний обмін газів у легенях. Вдихання кисню та видихання вуглекислого газу відбуваються швидше, що забезпечує вивільнення зайвих продуктів обміну газів.

Під час фізичного навантаження або тренувань інтенсивне дихання допомагає забезпечити додатковий потік кисню до м'язів, що сприяє їхньому ефективному функціонуванню. Такий тип дихання може призвести до зменшення рівня вуглекислого газу в крові. Це може бути корисно в умовах, коли організму потрібно вивести більше вуглекислого газу, наприклад, під час роботи м'язів під час фізичного зусилля.

Важливо враховувати, що при тривалому інтенсивному диханні може виникнути гіпервентиляція, що може призвести до зміни кислотно-лужного балансу в організмі. Гіпокапнія є станом, при якому рівень вуглекислого газу

(CO2) в крові знижується нижче нормальних значень. Це порушення може виникнути з ряду причин і впливати на фізіологічні процеси в організмі.

Гіповентиляція - стан, при якому об'єм подаваного легеням повітря недостатній для належного газообміну. Це може призвести до підвищення рівня вуглекислого газу в крові та зменшення рівня кисню.

Кількість кисню, яку клітини використовують, може значно варіюватися залежно від типу клітин, їхньої функції та активності. Проте, для більшості клітин кисень є необхідним для вироблення енергії в процесі клітинного дихання.

Основний процес, за якого клітини використовують кисень для отримання енергії, називається клітинним диханням. Клітинне дихання відбувається в мітохондріях клітин і включає три основні етапи: гліколіз, цикл Кребса та електронний транспортний ланцюг.

1.2. Легеневі об’єми та ємності

Легеневі об'єми та ємності визначають об'єм повітря, який може бути виміряний під час різних стадій дихального циклу.

Визначення легеневих об'ємів та ємностей є важливим аспектом для оцінки роботи легеневої системи та функції дихання. Ці вимірювання допомагають лікарям та іншим фахівцям у галузі респіраторної медицини отримати інформацію про стан легень та можуть бути використані для різних медичних і клінічних цілей.

Вимірювання легеневих об'ємів та ємностей є важливою частиною діагностики захворювань легень, таких як хронічна обструктивна хвороба легень, астма, бронхіт, фіброз легень та інші. У пацієнтів з хронічними захворюваннями легень вимірювання легеневих об'ємів може служити для визначення ефективності лікування та слідкування за прогресом хвороби. Вимірювання легеневих об'ємів дозволяє визначити, наскільки ефективно особа може вдихати та видихати повітря. Це допомагає в оцінці загальної функції дихання.

Для осіб із захворюваннями легень, включаючи тих, хто отримує лікування або реабілітацію, вимірювання легеневих об'ємів може бути використано для планування та моніторингу лікування. Для пацієнтів, які піддаються хірургічним втручанням, вимірювання легеневих об'ємів може допомогти визначити ризик для анестезії та операції.

Вимірювання легеневих об'ємів є важливою частиною досліджень, спрямованих на вивчення різних аспектів респіраторної функції.

Респіраторна функція організму включає в себе процеси дихання та газообміну, завдяки яким забезпечується нормальний обмін газів між організмом і зовнішнім середовищем.

Легенева вентиляція - це процес обміну повітрям між легенями та зовнішнім середовищем. Цей процес включає в себе два основних етапи: вдихання (інгаляція) та видихання (ексгаляція). Він забезпечує постачання

кисню до організму та видалення вуглекислого газу, який є відпрацьованим продуктом обміну газів в клітинах.

Розмір легеневої вентиляції, або об'єм легеневого обміну, може бути оцінений різними параметрами, які враховують об'єми та ємності легеневої системи.

Співвідношення між ємностями та обсягом наведено на рис. 1.3.

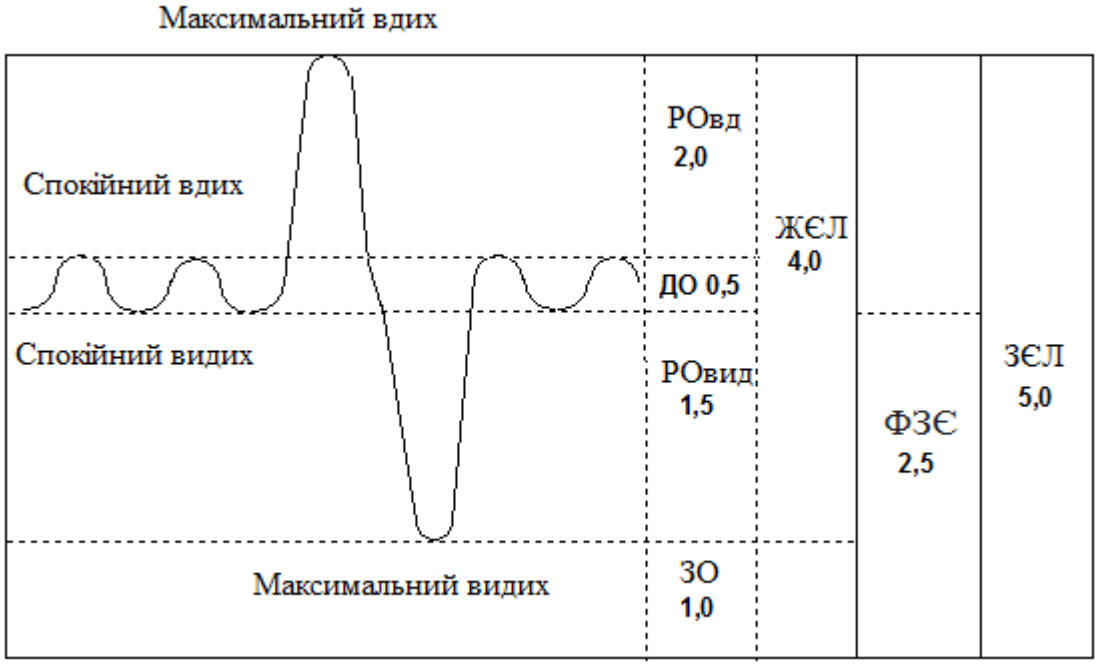

Рис. 1.3. Легеневі об’єми та ємності

Статичні легеневі об'єми вимірюють обсяги повітря у легенях в певних статичних або нерухомих умовах, без використання максимально можливих зусиль. Такі вимірювання часто використовуються для оцінки об'ємів, що визначаються при різних фазах дихання. Динамічні легеневі об'єми вимірюють обсяги повітря у легенях під час активних дихальних рухів, зазвичай за допомогою максимальних зусиль. Ці вимірювання дозволяють оцінити різні параметри, пов'язані з дихальною функцією та потужністю легеневого апарату [6, 10, 14].

Показники об'єму повітря у легенях та дихальних шляхах можуть залежати від ряду показників. Вони включають фізіологічні та анатомічні параметри, а також стан дихальної системи та активність особи. Основні показники, які впливають на об'єм повітря, включають:

1. Вік - об'єм повітря зазвичай зменшується з віком через природний процес старіння легеневої тканини та зміни в об'ємі інших дихальних параметрів.
2. Стать - у чоловіків, зазвичай, більший об'єм легеневого обміну порівняно з жінками. Це пов'язано з різницею в фізіології та анатомії.
3. Фізична активність - фізично активні люди можуть мати більший об'єм легеневого обміну, оскільки активність сприяє розвитку дихальної системи та підтримує її ефективність.
4. Здоров’я легень - стан легенів, включаючи наявність захворювань (наприклад, хронічних захворювань легень), може впливати на об'єм легеневого обміну.
5. Антропометричні параметри - розміри тіла, такі як ріст і маса, можуть впливати на об'єм легеневого обміну.
6. Генетичні фактори - генетичні особливості можуть визначати індивідуальні різниці в розмірах та потужності легеневого обміну.
7. Стан здоров’я - вірусні чи бактеріальні захворювання, алергії та інші стани можуть впливати на функціонування дихальної системи та об'єм легеневого обміну.

Дихальний об'єм - це об'єм повітря, який вдихається або видихається під час одного циклу дихання в стані спокою. Дихальний об'єм вимірюється у мілілітрах або літрах.

Цей параметр представляє собою кількість повітря, яке ви переміщаєте в легенях під час звичайного вдиху та видиху без додаткових зусиль. Визначення дихального об'єму є одним із ключових показників для оцінки легеневої функції та фізіології дихання.

Дихальний об'єм може залежати від умов виміру та контексту, в якому відбувається вимірювання. Визначення дихального об'єму може проводитися в різних умовах, і результати можуть варіюватися залежно від методів вимірювання та обраної методології.

Вимірювання дихального об'єму у стані спокою дозволяє отримати базовий показник, який відображає нормальні фізіологічні обсяги повітря при звичайних умовах. Під час фізичної активності дихальний об'єм може збільшитися, оскільки організму потрібно більше кисню для забезпечення м'язів під час фізичного навантаження.

Емоційний стан також може впливати на дихальний об'єм. Умови стресу чи страху можуть призвести до збільшення чи зменшення об'єму дихання. У лікарському контексті, такому як діагностика легеневих захворювань, дихальний об'єм може вимірюватися за допомогою спеціалізованих медичних приладів та тестів. Зміни в атмосферних умовах (тиск, температура, вологість) можуть впливати на об'єм легеневого обміну.

Визначаючи дихальний об’єм у нормі можна використати наступну формулу:

$$ДО = маса * 100.$$

Резервний об'єм вдиху (РОвд) – об'єм повітря, який можна додатково вдихнути після нормального вдиху. Вимірюється в мілілітрах або літрах. Резервний об'єм вдиху є одним із компонентів об'ємно-часових параметрів дихальної системи та може бути визначений за допомогою спірометрії.. Розмір РОвд становить 1,5 - 1,8 л.

$$РОвд = 0.34 * ЖЄЛ.$$

Резервний обсяг видиху (РОвид) - об'єм повітря, який можна видаляти з легенів після нормального видиху в стані спокою. Цей параметр є частиною об'ємно-часових параметрів дихальної системи та вимірюється в мілілітрах

або літрах. Може бути визначений за допомогою спірометрії. Розмір РОвид становить в середньому 1,0 - 1,4 л.

$$РОвид = 0.25 \times ЖЄЛ.$$

Ємність вдиху (Євд) - це об'єм повітря, який може бути вдихнутий після нормального видиху. Цей параметр є одним із ключових об'ємно-часових показників дихальної системи та використовується для характеристики об'ємів повітря, які можуть вміщуватися в легенях. У людини Євд становить у середньому 2,0 - 2,3 л.

$$Євд = ДО + РОвд$$

Функціональна залишкова ємність (ФЗЄ) – це об'єм повітря, який залишається в легенях після нормального видиху в стані спокою. Це один із ключових параметрів, що характеризує стан легеневої системи. Цей параметр вказує на об'єм повітря, який залишається в легенях після того, як вже відбувся нормальний видих. Функціональна залишкова ємність є важливою для збереження вентиляційного обміну газів в легенях, утримання стійкої оксигенізації та вуглекислоті.

$$ФЗЄ = ЗО + РОвид$$
$$ФЗЄ = ЗЕЛ - Евд$$

Життєва ємність легень (ЖЄЛ) - це об'єм повітря, який може бути видалений з легень після максимального вдиху, або видалений з легень після максимального видиху. Це важливий показник, який характеризує обсяг дихальних рухів та функцію легеневої системи. Життєва ємність легень

включає три компоненти: резервний об'єм вдиху, дихальний об'єм, резервний об'єм видиху:

$$ЖЄЛ = РОвд + ДО + РОвид$$

У чоловіків середнього віку життєва ємність легень зазвичай стаовить 3,5-5,0 л і більше. У жінок середнього віку життєва ємність легень зазвичай становить (3,0-4,0 л). Нормальні значення можна розрахувати за формулами:

$$ч: ЖЄЛ\ (мл) = (27{,}63 - 0{,}112 \times вік) \times зріст(см)$$
$$ж: ЖЄЛ\ (мл) = (21{,}78 - 0{,}101 \times вік) \times зріст(см)$$

Загальна ємність легень (ЗЄЛ) – це об'єм повітря в легенях, який включає в себе об'єм вдиху, резервний об'єм вдиху та резервний об'єм видиху. У нормі розраховують за формулами:

$$ч: ЗЄЛ = (36{,}2 - 0{,}06) \times вік \times зріст(см);$$
$$ж: ЗЄЛ = (28{,}6 - 0{,}06) \times вік \times зріст(см).$$

Залишковий об'єм (ЗО) – це об'єм повітря, який залишається в легенях після максимального видиху. Розмір залишкового об'єму дорівнює 1,0 -1,5 л.

$$ЗО = 0{,}25 \times ЗЄЛ$$

Вимірювання легеневих об'ємів та ємностей має кілька важливих клінічних та діагностичних застосувань:

- діагностика захворювань легень;
- оцінка функції легень;
- моніторинг лікування;
- оцінка ризику хвороби;

- планування реабілітації.

Існує кілька методів вимірювання легеневих об'ємів та ємностей, і вони можуть використовуватися як для діагностики, так і для моніторингу функції дихальної системи.

Статична спірометрія вимірює основні об'єми та ємності, такі як об'єм вдиху в стані спокою, резервний об'єм вдиху, резервний об'єм видиху. Динамічна спірометрія вимірює об'єми та ємності під час максимальних дихальних зусиль, включаючи життєву ємність легень.

Плетизмографія вимірює об'єм повітря у легенях шляхом визначення зміни об'єму тіла пацієнта. Це дозволяє оцінювати резервний об'єм легень та інші параметри.

Газообмінна мітка визначає різні параметри, такі як функціональна залишкова ємність та інші, використовуючи азот як газову мітку.

Магнітно-резонансна томографія та комп'ютерна томографія дозволяють отримати зображення структури легенів і визначити об'єми повітря та тканин.

Пневмокомп'ютерна томографія визначає залишковий об'єм та інші параметри за допомогою замкнутої камери.

Загальне вимірювання легеневих об'ємів та ємностей надає важливі клінічні дані. Ці дані допомагають лікарям зрозуміти стан дихальної системи та розробляти ефективні плани лікування для пацієнтів з захворюваннями легень.

1.3. Основні методи досліджень у пульмонології

1.3.1. Спірометрія

Це метод дослідження функціонального стану легень, який заснований на графічному відображенні змін їхнього об’єму при диханні (рис. 1.3). Спірометрія – нешкідливий та досить інформативний метод діагностики різних захворювань дихальної системи. З його допомогою визначають низку показників, аналіз яких дозволяє оцінити дихальну функцію легень пацієнта [19, 20, 23, 25].

1) частота дихання (ЧД) – число дихальних рухів за 1 хв. У здорових людей цей показник у середньому дорівнює 16-17;

2) дихальний об'єм (ДО) – об’єм повітря, що надходить у легені за один вдих. Цей показник настільки різниться в різних людей, що самостійного діагностичного значення немає. У чоловіків він коливається в межах від 300 до 1200 мл, у жінок від 250 до 800 мл;

3) хвилинний об’єм дихання (ХОД) – об’єм повітря, що надходить у легені за 1 хв. У середньому від 4 до 10 л;

4) життєва ємність легень (ЖЄЛ) – максимальний об'єм повітря, що видихається з легенів при глибокому видиху після максимально глибокого вдиху. У здорових людей становить від 2,5 до 7,5 л. Високі показники ЖЄЛ спостерігаються у плавців;

5) об'єм форсованого видиху за 1 с (ОФВ1) – максимальний об'єм повітря, що видихається з легенів за 1 с при форсованому (посиленому) видиху після максимально глибокого вдиху. Однозначно прийнятих за норму результатів цього виміру немає, оскільки все залежить від апаратури, статі та віку хворих;

6) індекс Тіффно (ІТ) – відношення ЖЄЛ до ОФВ1, виражене у відсотках.

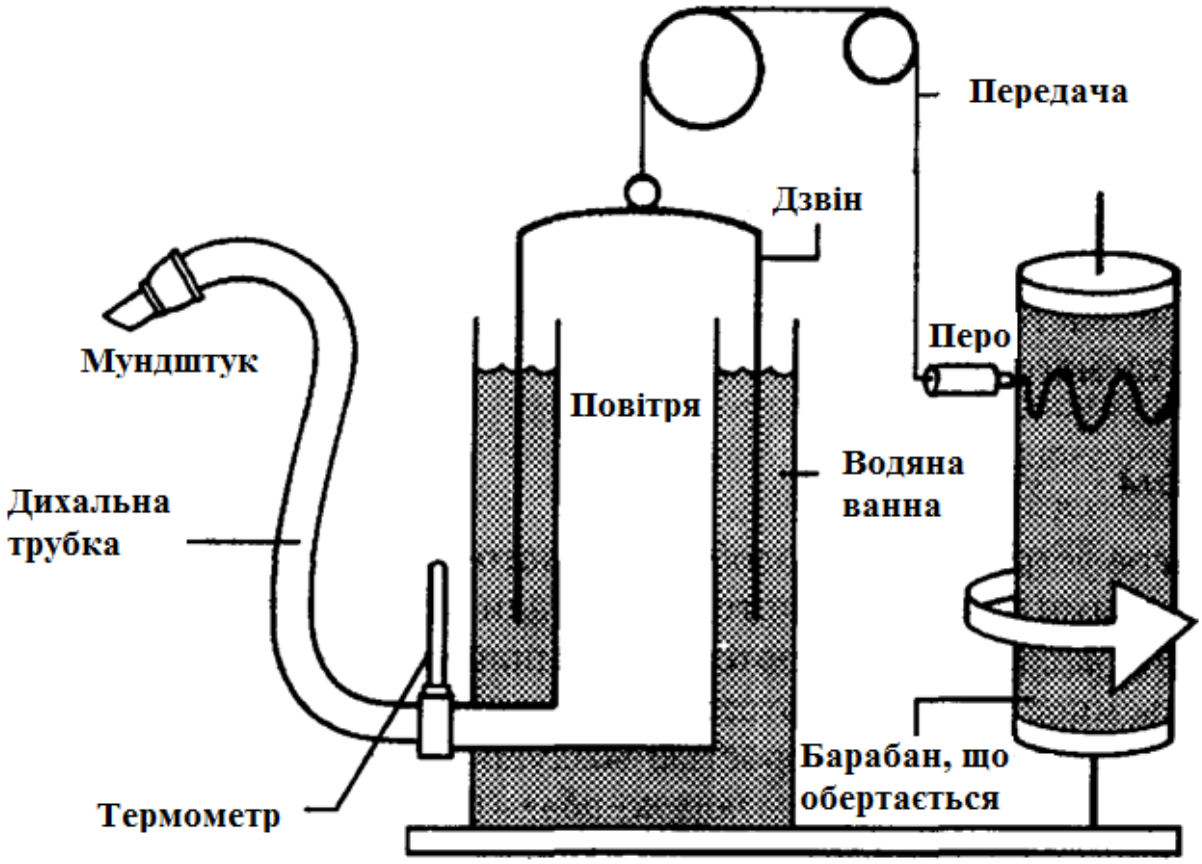

Рис. 1.4. Структура механічного спірометра

Для більшості показників існують так звані належні величини, тобто межі значень показників характерні для нормальної функції дихання. Звісно, показники функції дихання різні навіть у здорових людей. Тому, однозначно оцінити отримані дані, особливо в первинному дослідженні, досить складно. Положення полегшується, коли хворий має на руках результати попереднього спірографічного дослідження. У такому разі нові дані просто порівнюються з колишніми, і завдання лікаря правильно описати виявлені зміни.

Найбільшу роль у діагностиці ранніх стадій легеневих захворювань грають три показника: ЖЄЛ, ОФВ1 і ІТ. Інші показники вимірюють при необхідності, додатково до цих трьох.

1.3.2. Рентгенографія

Рентгенографія – це метод діагностики, що використовує рентгенівські промені для отримання зображення внутрішніх органів та структур. Основною ідеєю є те, що рентгенівські промені проходять через м'які тканини, такі як шкіра та м'язи, але поглиблюються в щільних тканинах, таких як кістки та органи.

Основні напрямки рентгенографічних досліджень:

- рентгенівські промені - високоенергетичне випромінювання, яке проймає тканини та реєструється рентгенівським фільмом чи цифровим сенсором;
- детектори - сучасні системи рентгенографії використовують цифрові детектори для запису отриманих зображень;
- рентгенівська трубка - генерує рентгенівські промені, які проникають через об'єкт, який досліджується;
- фільм або електронні системи - зображення може бути зафіксоване на рентгенівському фільмі чи електронному детекторі.

В рентгенографії можуть використовуватися різні проекції або напрямки, щоб отримати зображення певної області тіла. Деякі з основних проекцій включають пряму, бокову, діагональну, кутову та проекцію від кута.

1.3.3. Томографічні методи

Лінійна томографія легень, залежно від поставленого завдання, проводиться: у прямій, бічній та косих проекціях; з поздовжнім, косим та поперечним видом розмазування тіней; при вертикальному чи горизонтальному положенні пацієнта. Товщина шару, що виділяється при томографії залежить від кута повороту рентгенівської трубки. Чим він більший, тим менше товщина зрізу, що виділяється.

Основна увага при томографії приділяється правильному вибору глибини зрізу, що визначає діагностичну цінність одержуваних томограм. Для уточнення характеру інфільтрату в паренхімі легені слід проводити томографію у двох взаємно перпендикулярних проекціях.

Гортань і трахею томографують у двох взаємно перпендикулярних проекціях, головні та верхньодольові бронхи – у прямій та косій проекціях.

Оптимальною проекцією для нижньодольових бронхів легень є бічна, для середньочасткового та язичкового бронхів – коса. Бронхи верхівкового, заднього та переднього сегментів обох легень томографують у прямій та бічній проекціях. Оптимальною проекцією для отримання просвітів сегментарних бронхів середньої частки та язичкових сегментів є коса проекція з кутом повороту пацієнта на 50-55° щодо довжини осі тіла. Для отримання просвітів верхівкових сегментарних бронхів нижніх часток використовують бічну томографію. Томографія базальних сегментів проводиться у прямій та бічній проекціях. У цих проекціях одержують відображення просвіти базальних бронхів нижньої частки.

Зонографія – більш товстий шар, що виділяється. Її краще застосовувати для уточнення характеру дисемінованих процесів легень, деталізації рентгенологічної картини фіброзних та склеротичних змін, легеневого малюнка та коріння легень.

Із сучасних спеціальних методів отримання зображення у діагностиці захворювань легень та ЛОР-органів методом вибору є комп'ютерна томографія (КТ).

Цифрова обчислювальна томографія - це процес застосування рентгенівської томографії, яке використовує комп'ютер для обробки і обчислення тривимірних зображень внутрішніх структур тіла. СТ дозволяє отримувати детальні та крос-секційні зображення, що робить його вельми корисним для діагностики та оцінки різних медичних станів.

Універсальність цифрової обчислювальної томографії базується на застосуванні цілого комплексу методів. Рентгенівські промені використовуються для проникнення через тканини та отримання зображень.

Комп’ютерна обробка - інформація, отримана від детекторів після проходження променів, обробляється комп'ютером для створення детальних тривимірних зображень.

Надаються крос-секційні (поперечні) зображення внутрішніх структур, що дозволяє лікарям вивчати деталі певного шару тіла.

Має високу роздільну здатність, що дозволяє отримати докладні зображення, що важливо для точної діагностики.

Діагностика різних захворювань - використовується для діагностики і вивчення різних станів, таких як пухлини, травми, інфекції, високий кров'яний тиск, хвороби судин та інші.

При дослідженні грудної порожнини цифрова обчислювальна томографія може використовуватися для отримання детальних зображень внутрішніх структур грудей та оцінки різних патологій. Таке дослідження грудної порожнини може бути здійснено за різних умов та для різної мети. Наприклад, виявлення пухлин, діагностики захворювання легень, вивчення серцево-судинної системи та ін.

Під час проведення дослідження грудної порожнини пацієнт зазвичай лежить на столі сканера, який обертається навколо тіла, збираючи рядки даних. Отримані дані комп'ютер оброблює для створення детального тривимірного зображення внутрішніх органів та структур. Томографічне дослідження у багатьох випадках здатне замінити інвазивне дослідження – ангіографію.

1.4. Аналіз ринку медичних інформаційних систем

Світовий ринок медичних інформаційних систем (МІС) на даний час представлений чималою кількістю учасників та рішень. Перше дослідження

ринку всього спектра програмного забезпечення (ПЗ) для медицини було проведено у 1998 р. У той час найбільшого поширення набули програми для автоматизації фінансових та адміністративних служб лікувально-профілактичних установ (ЛПЗ), а системи, призначені безпосередньо для лікувально-діагностичного процесу, становили лише 9,6 % всього ПЗ для медицини. У 2000 р. ситуація дещо покращилася, проте суттєвих змін відтоді не відбулося. Така стабільність пов'язана з тим, що розробка МІС займає тривалий термін (не менше двох-трьох років, середня тривалість розробки системи з широкими функціональними можливостями приблизно становить 7 років), тому кількість нових систем у цій сфері зростає повільно.

Основним недоліком більшості МІС, представлених на ринку в 2000-2002 рр., була їх вузька спеціалізація. Найбільш популярними були системи відділень стоматології, офтальмології, рентгенології, анестезіології, реаніматології та інших областей. І зараз є кілька програм, призначених для автоматизації лише реєстратури, доцільність застосування яких є сумнівною у відриві від автоматизації клінічних відділень, відділу статистики та інших основних служб ЛПЗ [24].

Сучасна МІС має бути універсальною і підтримувати різні види ЛПЗ, а не спеціалізуватися, наприклад, лише на автоматизації окремого підрозділу клініки [21].

Предметом цього аналізу є стан ринку лише основних МІС, спрямованих на комплексну автоматизацію ЛПЗ та об'єднуючих кілька взаємопов'язаних підсистем, таких, як електронна історія хвороби (ЕІБ) або амбулаторна карта пацієнта, розклад робочого часу персоналу, бухгалтерія, статистика та ін. Майже 80% з них можуть застосовуватися в багатопрофільних стаціонарах та поліклініках, у санаторно-курортних установах.

Функціональні можливості модулів, включених до складу інформаційних систем, обмежуються веденням бази даних, формуванням направленням на дослідження, виписок і можуть бути інтегровані з

програмним забезпеченням, що поставляється у складі діагностичного обладнання. Загалом виконувані ними функції зводяться до наступного:

- введення, редагування та пошук даних пацієнтів;
- робота з даними, отриманими під час обстежень;
- перегляд графіків отриманих спірограм;
- побудова теоретичних кривих для ФЖЕЛ;
- таблиці для перегляду даних вимірів, належних величин та відхилень;
- висновок позначень для досліджуваних величин російською та англійською мовами;
- автоматична видача попереднього висновку;
- попередній перегляд звіту з результатами обстежень, що включає дані пацієнта, результати розрахунку належних і вимірюваних величин, відхилень, а також графіків та висновків;
- друк детального звіту з результатами обстежень.

Таким чином, до недоліків існуючих систем можна віднести те, що до складу БД вибірково включаються лише деякі результати обстежень.

У 2000 роках 47% МІС використовували локальні (настільні) БД, причому у переважній більшості випадків це були різновиди dBase. Такий підхід характерний для початкового періоду розробок програмного забезпечення для медицини та створення вузькоспеціалізованих продуктів. З кожним роком кількість вітчизняних систем на основі настільних БД зменшується, 2003-го, за нашими даними, ця цифра становила вже лише 4%. На сьогодні лише деякі розробники використовують dBase як формат зберігання даних та FoxPro як програмне середовище. Враховуючи, що Microsoft вже неодноразово робила заяви про плани припинення розвитку лінійки FoxPro, цей напрямок можна вважати не перспективним.

Деякі програмні продукти використовують власний формат БД, нерідко такі формати застосовуються в електронних фармакологічних довідниках. Нині над ринком є МІС, побудована навіть у власній СУБД архітектури "клієнт - сервер": e-Hospital.

Переважна більшість МІС побудована в архітектурі "клієнт-сервер". Практичним досвідом доведено неминучість такого рішення для створення комплексної МІС, оскільки файл-сервер здатний підтримувати лише до 10 робочих станцій та невеликий обсяг бази даних. Крім того, значна частина існуючих вимог до МІС вже реалізована в промислових СУБД, що дозволяє суттєво скоротити час на створення системи.

Під час розробки МІС переважно застосовуються такі СУБД: Microsoft SQL Server (версія 7.0 чи 2000) – 29,4%, Oracle – 17,6%, BorlandInterbase Server – 5,8%, Cache – 11,7%, Lotus Notes/Domino – 11,8%.

Усі СУБД поділяються на два принципово різних види: реляційні (РБД) і постреляційні (об'єктно-орієнтовані – ООБД). Серед МІС перевага РБД не така безумовна – 75%. Інші 25% займають постреляційні СУБД. При цьому в цю категорію ми включили Lotus Notes/Domino, яку умовно можна назвати постреляційною – це скоріше документно-орієнтована платформа для групової роботи. Lotus Notes/Domino і Cache зайняли паритетні позиції – обом належить по 50% сегменту стріляння СУБД.

Аналіз літературних джерел щодо сучасного стану ринку медичних інформаційних систем (МІС) показує наступні напрямки розвитку:

1. Цифрова трансформація в охороні здоров’я: багато країн і організацій вкладають ресурси в цифрові технології для поліпшення доступу до медичних послуг, підвищення ефективності та зменшення помилок в лікуванні.
2. Електронна медична картка: запровадження електронних медичних карток та інших електронних систем управління медичною інформацією сприяє зберіганню та обміну даними між медичними установами.
3. Аналітика та штучний інтелект (ШІ): застосування аналітики даних та ШІ в медичних інформаційних системах може допомагати виявляти тенденції, прогнозувати захворювання та підтримувати прийняття рішень у медицині.

4. Збільшення кількості мобільних додатків для здоров’я: мобільні додатки стають все більш популярними для ведення особистого здоров'я, моніторингу фізичної активності, дієти та інших аспектів здоров'я.
5. Безпека даних і конфіденційність: збільшення кількості цифрових даних вимагає підвищеної уваги до питань безпеки даних та забезпечення конфіденційності пацієнтів.
6. Стандартизація даних: впровадження стандартів для обміну та зберігання медичної інформації допомагає різним системам взаємодіяти та обмінюватися даними.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1. Структура бази даних

У попередньому розділі ми розглянули специфічну інформацію щодо інструментальних досліджень легень. Ми будемо використовувати результати спірометричних і рентгенівських досліджень у даному проекті.

Для збору та зберігання інформації нами обрана реляційна база даних – найпоширеніший тип, який має низку переваг, дозволяє створювати інтуїтивно зрозумілі зв'язки між даними.

Структура таблиць наведена у додатку А і включає чотири таблиці. Головна таблиця **patient** містить загальні паспортні дані про пацієнта: П.І.Б., стать (чоловіча або жіноча), дату народження, адресу та телефон. Ключове поле **id** є ідентифікатором запису та автоматично збільшується на 1 після додавання нового рядка до таблиці. З метою зменшення обсягу даних, що зберігаються, ми обмежуємо число символів для зберігання деякої текстової інформації: 30 символів для П.І.Б. (для кожного поля, varchar (30)) і один символ для статі (varchar (1)).

Ми розробляємо онлайн-модуль, тому звернулися до послуг безкоштовного хостингу, що надає місце та підтримку php, MySQL: www.000webhost.com.

Було пройдено реєстрацію безкоштовного облікового запису на сервері, як доменне ім'я, доступ до сервісу здійснюється за посиланням:

https://diplomaproject.000webhostapp.com/index.php

Хостинг має панель управління, одні із розділом якого є інструменти для роботи з базами даних.

Список полів головної таблиці, їх розмір та тип представлені на рис. 2.1. Малюнок відображає частину інтерфейсу робочого вікна для роботи з базами даних.

Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**id**	int(11)			No	*None*
☐	2	**first_n**	varchar(30)	utf8_unicode_ci		No	*None*
☐	3	**last_n**	varchar(30)	utf8_unicode_ci		No	*None*
☐	4	**year_b**	int(11)			No	*None*
☐	5	**sex**	varchar(1)	utf8_unicode_ci		No	*None*
☐	6	**address**	varchar(100)	utf8_unicode_ci		No	*None*
☐	7	**tel**	varchar(15)	utf8_unicode_ci		No	*None*

Рис. 2.1. Структура таблиці *patient*

Друга таблиця (підлегла) **visit** містить дані про візит до лікаря. Її структура відображена на рис. 2.2. Ключове поле **id** є унікальним ідентифікатором. Для зв'язку таблиць служить поле **id_p**, яке записується значення поля **id** з головної таблиці **patient**. Т.к. пацієнт може звертатися до лікаря більше одного разу, ми використовуємо тип відношення «один-багатьом».

Поля **spiro** і **xray** необхідні, щоб зробити позначку, чи проводилися спірометричні та рентгенівські дослідження. П.І.Б. лікаря (поле **doctor**) записуються в цю саму таблицю, як і лікарський висновок (поле **conclusion**). Число символів неможливо зафіксувати, тому ми вибрали тип тексту з невизначеною розмірністю. Два поля **mass**, **height** необхідні в подальшому, щоб зробити розрахунок основних спірометричних показників у нормі з урахуванням антропометричних даних та віку пацієнта.

	#	Name	Type	Collation	Attributes	Null	Default	Comments
☐	1	**id**	int(11)			No	*None*	
☐	2	**id_p**	int(11)			No	*None*	зв'язок id patient
☐	3	**date**	date			No	*None*	
☐	4	**doctor**	varchar(30)	utf8_unicode_ci		No	*None*	
☐	5	**spiro**	tinyint(1)			No	*None*	1 - є; 0 - немає
☐	6	**xray**	tinyint(1)			No	*None*	1 - є; 0 - немає
☐	7	**conclussion**	text	utf8_unicode_ci		No	*None*	
☐	8	**mass**	double			No	*None*	
☐	9	**height**	double			No	*None*	

Рис. 2.2. Структура таблиці ***visit***

Таблиця **spiro** призначена для зберігання спірометричних даних: дихальний об'єм (**TD**), загальна ємність легень (**TLC**), резервний об'єм вдиху (**IRV**), резервний об'єм видиху (**ERV**), ємність вдиху (**IV**), функціональна залишкова ємність (**FRC**), життєва ємність легень (**VC**), залишковий об'єм (**RC**). Її структура та опис полів показані на рис. 2.3. **spiro** є підпорядкованою таблиці **visit**; тип відношення «один-багатьом» і реалізований за схемою **id** (з **visit**) →**id_v** (із **spiro**). Для числових даних вибрано тип *float*.

Четверта таблиця **xray** використовується для зберігання зображень, а точніше посилань на графічні файли (**ximage**), які завантажуються у виділену папку, що зберігається на сервері. Таким чином, таблиця має чотири поля (рис. 2.4), в одне з яких (**resume**) записується висновок по ренгенівському зображенню. Таблиця підпорядкована **visit**.

	#	Name	Type	Collation	Attributes	Null	Default	Comments
☐	1	**id**	int(11)			No	*None*	
☐	2	**id_v**	int(11)			No	*None*	зв'язок з id visit
☐	3	**TD**	double			No	*None*	Дихальний об'єм
☐	4	**TLC**	double			No	*None*	Загальна емність легень
☐	5	**IRV**	double			No	*None*	Резервний об'єм вдиху
☐	6	**ERV**	double			No	*None*	Резервний об'єм видиху
☐	7	**IC**	double			No	*None*	Емність вдиху
☐	8	**FRC**	double			No	*None*	Функ. емність легень
☐	9	**VC**	double			No	*None*	ЖЄЛ
☐	10	**RC**	double			No	*None*	Залишковий об'єм

Рис. 2.3. Структура таблиці ***spiro***

Таблиця **xray** підпорядковується таблиці **visit**. Тип відносини «один-багатьом», здійснюється по полях **id** (із **visit**) →**id_v** (із **xray**).

	#	Name	Type	Collation	Attributes	Null	Default	Comments
☐	1	**id**	int(11)			No	*None*	
☐	2	**id_v**	int(11)			No	*None*	зв'язок з id visit
☐	3	**ximage**	text	utf8_unicode_ci		No	*None*	посилання на файл
☐	4	**resume**	text	utf8_unicode_ci		No	*None*	висновок по зображенню

Рис. 2.4 Структура таблиці ***xray***

2.2. Взаємодія модулів СУБД

СУБД є набором модулів, що взаємодіють один з одним, і забезпечують збір, зберігання, доступ та маніпулювання даними. Існують також вівуальні компоненти, які об'єднані в інтерфейсі користувача. Взаємодія з фізичними таблицями бази даних також здійснюється за допомогою інтерфейсу користувача. У цілому нині, СУБД реалізована у вигляді виконуючих php-файлів, кожен із яких має призначення.

Повна схема взаємодії між модулями СУБД представлена в додатку Б. Доступ до даних надається через процес аутотентифікації (пара «логін-пароль»).

Головна сторінка (**index.php**) містить три незалежні фрейми: верхній (**head.php**), лівий (**menu.php**) та вікно для відображення основного вмісту (**content.php**). Останній фрейм - динамічний, в нього вивантажуються файли в залежності від дій користувача: вибір меню, заповнення форм і т.д.

Головне меню має три основні пункти:

- **Пошук** існуючого електронного запису;
- **Створення** нового електронного запису про пацієнта;
- **Видалення** електронного запису про пацієнта.

Коли користувач вибирає пункт **Пошук**, відкривається форма для введення критеріїв пошуку (**search_form.tpl**). Після закінчення заповнення полів, файл проходить перенаправлення до **search.php**, в якому здійснюється пошук даних у таблиці **patient**. Якщо запису не знайдено, користувач буде перенаправлений на початкову сторінку **content.php**.

Якщо запис існує, керування передається файлу **patrec.php**, який відповідає за відображення вмісту запису. **id** знайденого запису визначає співвідношення між записами в таблиці **visit**, тобто, служить другою частиною запиту. Далі ми викликаємо **visit.php**, щоб отримати дані про візити до лікаря та надіслати запити для пошуку записів про дослідження (**spiro.php** і **xray.php**).

Щоб створити новий запис, необхідно виконати перенаправлення на форму, що містить поля для введення даних (**new_pat_form.php**). Після їх заповнення керування передається файлу **new_pat.php** для передачі і запису даних в базу. На наступному кроці нам пропонують ввести дані про результати медичних досліджень (**new_spiro.php** і **new_xray.php**).

Останній пункт у меню викликає файл **delete_pat.php**, який видаляє поточний запис.

2.3. Інтерфейс СУБД

Оскільки програма орієнтована на онлайн-режим, ми будемо дотримуватися принципу фреймової розмітки сторінки з динамічним вмістом сторінок. Стандартний варіант фреймової структури інтерфейсу використовується нами та показаний на малюнку 2.5.

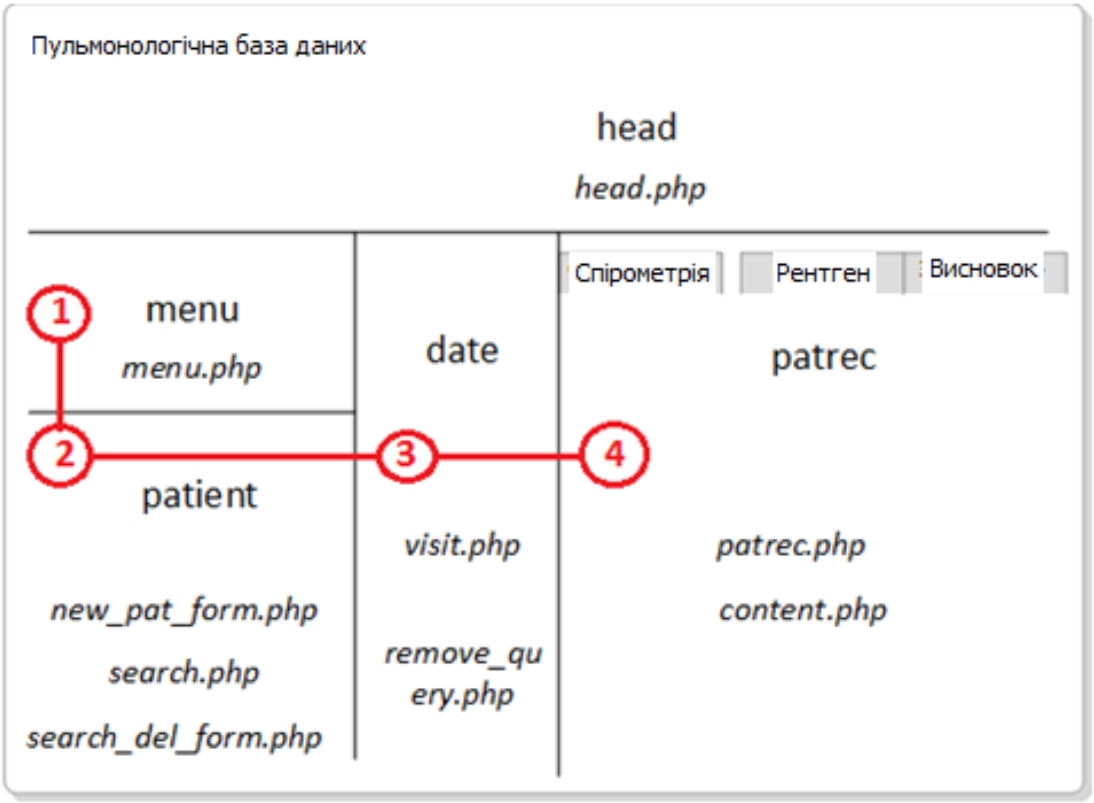

Рис. 2.5. Розмітка сторінки

Фрейм **patient** має динамічний вміст, так як сюди вивантажуються результати виконання файлів **new_pat_form.php** (якщо ми хочемо створити новий запис), **search.php** (якщо будемо здійснювати пошук записів), **search_del_form.php** (для видалення поточного запису). Таким чином, цей кадр контролюється пунктами меню.

Вибраний запис дозволяє відобразити пов'язану інформацію у фреймі **visit**, який містить дати відвідувань у вигляді таблиці. За натисканням на дату результати досліджень будуть знайдені і виведені у кадр **patrec**.

Таким чином, логічна послідовність, яка позначена цифрами на рис. 2.5, визначає файли, що підвантажуються в кожен фрейм, і повністю відповідає зв'язкам між таблицями бази даних.

Логічні блоки модуля тісно пов'язані з його інтерфейсом. Структура модуля наведена в додатку В. Вона включає блоки для авторизації користувача, що передбачає різні рівні доступу до даних, наприклад, для лікарів. Авторизація відбувається за допомогою пари «логін-пароль». Блок реєстрації нових користувачів контролюється адміністратором системи. Дані про користувачів повинні зберігатися в окремій базі даних.

Друга база даних призначена для зберігання пульмонологічних даних пацієнтів. Взаємодія з нею здійснюється за допомогою блоків введення та пошуку даних. Блок розрахунку показників необхідний, щоб обчислити нормальні значення спірометричних даних залежно від віку, статі та ваги пацієнта. Блок оцінки та відображення даних фактично є інтерфейсом користувача.

Розглянута структура визначає логічний зв'язок між його елементами і допомагає визначити їх функціональне призначення.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Програмні засоби

База даних та СУБД реалізовані у вигляді онлайн-системи. Для цього був використаний безкоштовний онлайн-сервіс www.000webhost.com, який забезпечує підтримку MySQL, php My Admin, php і java: Apache 2.2.19 version, php 5.3 version, MySQL 5.1 version.

Php був обраний з низки причин. Перш за все, він широко використовується для вирішення схожих завдань, має функції для роботи з базами даних, текстом, легко поєднується з HTML, а також дозволяє впроваджувати html-теги безпосередньо у виконуваний php-код.

MySQL - джерело з відкритим кодом, який дозволяє розробляти і обслуговувати реляційні бази даних. Основною його перевагою є кросплатформність, сумісність із більшістю мов програмування, інтуїтивно зрозумілий синтаксис. Це мова для маніпуляції даними, стандартними командами якої є ADD, DROP, INSERT та UPDATE.

MySQL може бути використаний безліччю програм, але найширше використовується web-серверами. Веб-сайти, які використовують MySQL, можуть включати інформацію, яка виводиться з бази даних у веб-сторінку. Тому вони називаються динамічними. Таким чином, будь-яка веб-сторінка може мати постійний та змінний вміст, що вивантажується у відповідь на запит користувача.

Багато веб-сайтів, контент яких вивантажується з бази даних, використовують скрипти, подібні до php. Команди MySQL можуть бути вбудовані в php-скрипт, що забезпечує доступ до бази та маніпуляції з даними. MySQL і php – джерела з відкритим кодом, що безкоштовно розповсюджуються, що робить їх спільне використання дуже популярним.

3.2. Створення бази даних

У пункті 2.1 було описано структуру бази даних. Процес фізичного створення бази показано на рис. 3.1. Панель керування хостингом має пункт меню для роботи з базами даних; ми задаємо ім'я бази даних, ім'я користувача та пароль.

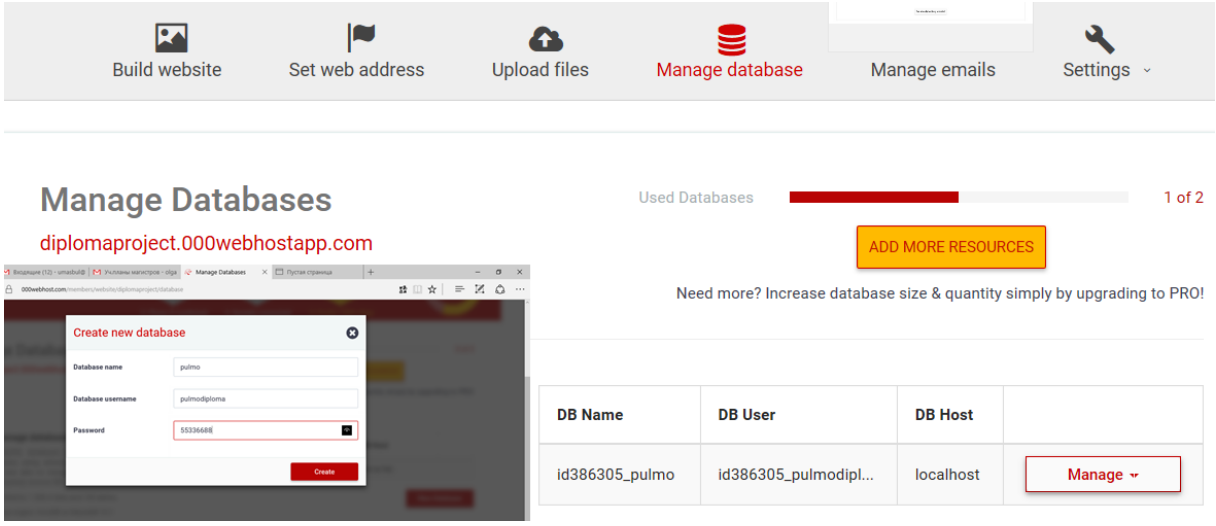

Рис. 3.1. Створення бази даних

Ім'я бази даних (DBName), ім'я користувача (DBUser) та адреса хоста (DBHost) надалі будуть використовуватися для встановлення з'єднання з базою даних та доступу до даних.

3.3. Головна сторінка

Головна сторінка (index.php) має фреймову структуру (рис. 2.5) та реалізується за допомогою наступного коду:

```
<FRAMESET ROWS="10%,80%" FRAMEBORDER="YES" BORDER="2">

<frame src="head.php" name="head">
```

```
<FRAMESET COLS="25%,75%" FRAMEBORDER="YES" BORDER="2">
<FRAMESET ROWS="25%,75%" FRAMEBORDER="YES" BORDER="2">
<frame src="menu.php" name="menu">
<frame src="" name="patient">
</FRAMESET>
<FRAMESET COLS="25%,75%" FRAMEBORDER="YES" BORDER="2">
<frame src="" name="date">
<frame src="" name="patrec">
</FRAMESET>
</FRAMESET>
</FRAMESET>
```

3.4. Меню

Файл **menu.php** забезпечує перехід за посиланням та запуск відповідного файлу (рис. 3.2). При цьому той чи інший файл відкривається у кадрі **patient**. Якщо користувач вибирає пошук, відображатиметься форма, збережена у файлі **search_form.tpl**. У ній передбачені поля для введення даних згідно з критеріями пошуку.

Пункт меню **нова карта** завантажує файл **new_pat_form.php**, де користувач вводить дані, які потім передаються для запису в базу.

Пункт **видалити карту** відкриває файл **search_del.php**, що дозволяє виконати пошук карток за заданим критерієм, а потім видалити вибрану користувачем історію хвороби.

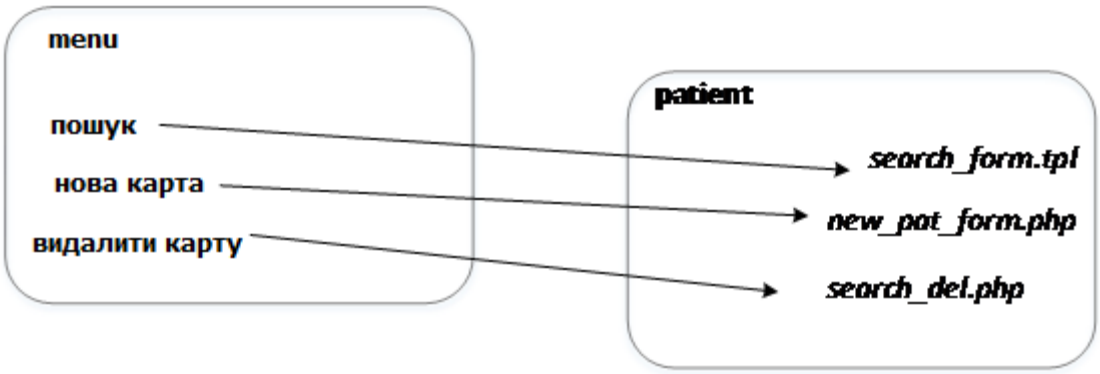

Рис. 3.2. Схема меню

Код, що реалізує меню, дуже компактний і наводиться нижче:

```
<h1>меню</h1>

<a href='search_form.php' target='patient'>пошук</a><br>

<a href='new_pat_form.php' target='patient'>новакартка</a><br>

<a href='search_del.php'>видалитикартку</a><br>
```

3.5. Пошук історії хвороби

3.5.1. Пошук у таблиці 'patient'

Щоб виконати пошук, потрібно задати критерії. Ми вибрали декілька критеріїв для пошуку: ім'я, прізвище, стать та рік народження. Форма для введення реалізована на HTML, поля для введення – за допомогою тегів **<input>**, радіокнопок для вибору поля та кнопки для надсилання даних (рис. 3.3).

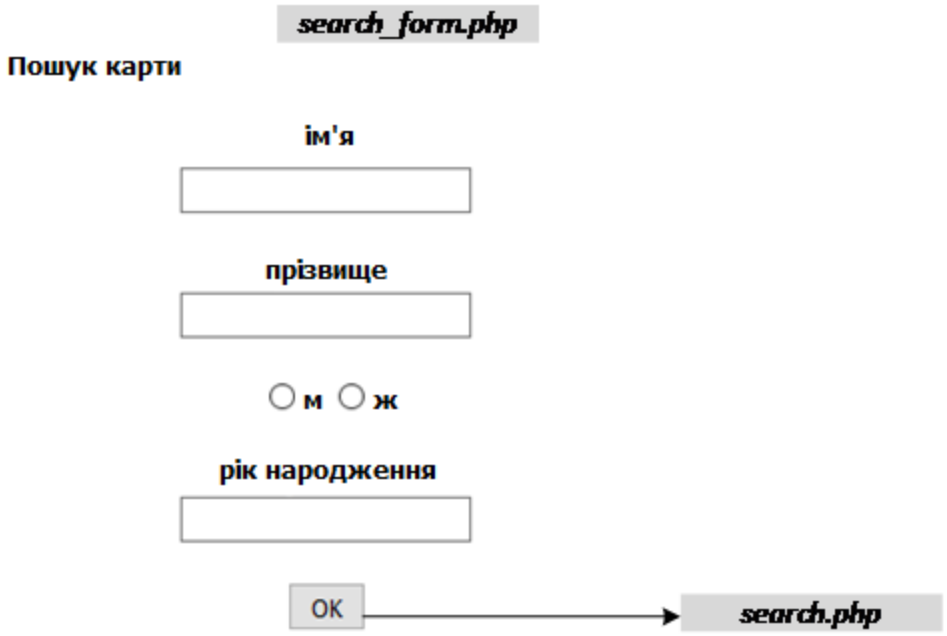

Рис. 3.3. Файл **search_form.php**

Код форми наведено нижче:

```
<h4>Пошуккарти</h4>

<form action='search.php' method='GET'>

<div align='center'>

имя<br><input name='first_n' type='text'>

<br><br>

прізвище<br><input name='last_n' type='text'/>



<br><br>

<input type='radio' name='sex' value='male'>м

<input type='radio' name='sex' value='female'>ж<br>

<br>

рікнародження<br><input type='text' name='year_b'>
```

```
<br><br>

<div align='center'><input type='submit' value='OK'>

</div>

</form>
```

Кожен елемент форми має ім'я, з якого дані полів будуть передаватися змінним наступного файлу **search.php**, де і реалізується пошуковий запит. Дані передаються за протоколом GET. Передача керування файлом **search.php** проводиться після натискання на кнопку OK.

Розглянемо реалізацію пошукового запиту у файлі **search.php**. На початку файлу ми інкапсулюємо файл **connect_db.php**, який містить код для встановлення з'єднання з базою даних, використовуючи ім'я бази даних, ім'я користувача та адресу хоста (всі ці параметри показані на рис. 3.1 та описані вище).

Пізні версії php мають нові функції для встановлення постійного з'єднання з сервером **(mysqli_connect())** та базою даних **(mysqli_select_db())**. Ми неодноразово встановлюватимемо з'єднання з базою з різних файлів, тому вирішили помістити код, що виконує цю функцію, в окремий файл **connect_db.php**, а потім інкапсулювати його в потрібному місці будь-якого файлу. Це одна з переваг php.

Також ми використовуємо ще один прийом – сесії. Будь-яка змінна може бути розміщена в сесію для зберігання, а потім викликана з іншого файлу. Це один із механізмів передачі даних між файлами.

Код файлу connect_db.php наведено нижче.

```
<?php
session_start();

//Змінні

$dbname='a5948319_pulmo';
```

```
$dbuser='a5948319_pulmo';
$dbhost='mysql12.000webhost.com';
$dbpassword='*****';

//з’єднання з сервером

$link=mysqli_connect($dbhost,$dbuser,$dbpassword,$dbname);

if(!$link) {
die("Помилкавходуваккуант".mysql_error());
}
$_SESSION['link']=$link;
?>
```

Щоб передати змінні між файлами, ми використовуємо масив глобальних змінних **$_SESSION**. Для масиву виділяється область пам'яті, коли відкривається сесія **(session_start())**, що рекомендують робити на початку файлу. Якщо з'єднання з базою даних пройшло успішно, функція **mysqli_connect** повертає ідентифікатор, значення якого присвоюється змінної **$link**. Нульове значення означає успішне з'єднання з базою даних. $link надалі може знадобитися для перевірки наявності з'єднання з базою, тому ми зберігаємо його в **$_SESSION[‘link’]**.

Ми впроваджуємо файл **connect_db.php**, потім задаємо ім'я таблиці в перемінній **$table** для запиту. Дані з форми отримуємо з масиву глобальних змінних **$_GET[‘ ‘]**:

```
<?php
include('connect_db.php');
$table='patient';

//Get from the form
$first_n=$_GET['first_n'];
$last_n=$_GET['last_n'];
```

```
$year_b=$_GET['year_b'];
$sex=$_GET['sex'];
```

Пошуковий запит залежить від обраних користувачем критеріїв. Тому в кожному випадку необхідно написати окремий рядок запиту SQL. Щоб розпізнати результат вибору критеріїв користувачем, ми введемо зміну **$flags**, яка є сумою допоміжних змінних:

```
$flags=$flag_FN+$flag_LN+$flag_YB+$flag_S.
```

Кожна зміна відповідає за один критерій, а при виборі користувачем будь-якого критерію відповідної змінної надається суворо певне значення. Щоб забезпечити унікальність при виборі ми пропонуємо значення змінних, вибірково представлені в табл. 3.1.

Таблиця 3.1.

Значення змінної **$flags** для двох параметрів

Поле/змінна	first_n $flag_FN=1	last_n $flag_LN=2	sex $flag_S=4	year_b $flag_YB=8
first_n $flag_FN=1	**1**	3	5	9
last_n $flag_LN=2	3	**2**	6	10
Sex $flag_S=4	5	6	**4**	12
year_b $flag_YB=8	9	10	12	**8**

Програмно змінним задаються відповідні значення, якщо поле форми не є порожнім, що демонструє наведений нижче фрагмент коду:

```
if(!empty($first_n))
     {
        $flag_FN=1;
}
```

```
if(!empty($last_n))
     {
        $flag_LN=2;
     }

if(!empty($year_b))
     {
        $flag_YB=4;
     }

if(!empty($sex))
     {
        $flag_S=8;
}
```

Наступним кроком є обчислення суми всіх змінних:

```
$flags=$flag_FN+$flag_LN+$flag_YB+$flag_S;
```

Значення **$flags** визначає, який пошуковий запит необхідно надіслати. Для цього ми застосовуємо конструкцію **switch – case**, де для кожного значення **$flags** записуємо пошуковий запит до рядкової змінної **$sql**.

Продемонструємо цю процедуру на наступному фрагменті коду, в якому коментарі пояснюють, для якого критерію пошуку формується запит:

```
switch ($flags) {
        ……

//Пошук по прізвищу, поле last_n
case 2:
$sql=”SELECT * FROM `$table` WHERE `last_n` like’$last_n%’”;
break;
…
//Пошук по всім критеріям
```

```
     case 15:

$sql=”SELECT * FROM `$table` WHERE `first_n` like ‘$first_n%’
AND `last_n` like ‘$last_n%’ AND `sex` = ‘$sex’ AND `year_b`
like ‘$year_b%’”;
               break;
   }
$rs=mysqli_query($link,$sql);
echo "<h3>Знайденокарт:".mysqli_num_rows($rs)."</h3><br>";
```

Число знайдених записів у головній таблиці та їх список буде відображати у фреймі **patient** як гіперпосилання. Коли користувач натисне гіперпосилання, що його цікавить, буде сформований новий запит в таблицю **visit**, щоб знайти і відобразити всі візити до лікаря для поточного пацієнта. У цьому має виконуватися умова, що закладена у структурі реляційної бази даних таблиць **patient-visit: id=id_p**.

Число знайдених записів визначаємо за допомогою функції **mysqli_num_rows()**.

Щоб маніпулювати полями зі знайденого запиту, скористаємося функцією **mysqli_fetch_array()**, яка заносить поля, що повертаються запитом до асоціативного масиву, коміркам якого присвоюються ті ж імена, що й поля фізичної таблиці. Це дуже зручно, оскільки автоматично повертає імена полів та попереджає появу помилок.

Поля для кожного знайденого запису запишемо до рядкової змінної **$row**; порядок полів визначаємо самі. Так як кількість знайдених записів, що повертаються запитом, заздалегідь невідома, виводити їх будемо в циклі **while**:

```
     while($patient=mysqli_fetch_array($rs))
     {
     $row=$patient['id']."                    ".$patient['first_n']."
".$patient['last']."                              ".$patient['sex']."
```

```
".$patient['year_b']."                    ".$patient['address']."
".$patient['tel'];


    echo      "<a      href='visit.php?idpat='.$patient['id']."'
Target='patrec' >".$row."</a><br>";
    }
    ?>
```

Кожна карта виводиться як гіперпосилання. Коли користувач натискає на одній з них, одночасно з переходом до файлу **visit.php** передається **id** пацієнта: **"<ahref='visit.php?idpat=".$patient['id']."'** Справа в тому, що php-скрипт виконується на стороні сервера, а користувачеві повертається результат виконання. Ця конструкція дозволяє передати за допомогою змінного **idpat** ідентифікатор запису **$patient['id']**. Надалі ідентифікатор запросить файл **patrec.php** через змінну **$_GET['idpat']**. Результат пошуку показано на рис. 3.4.

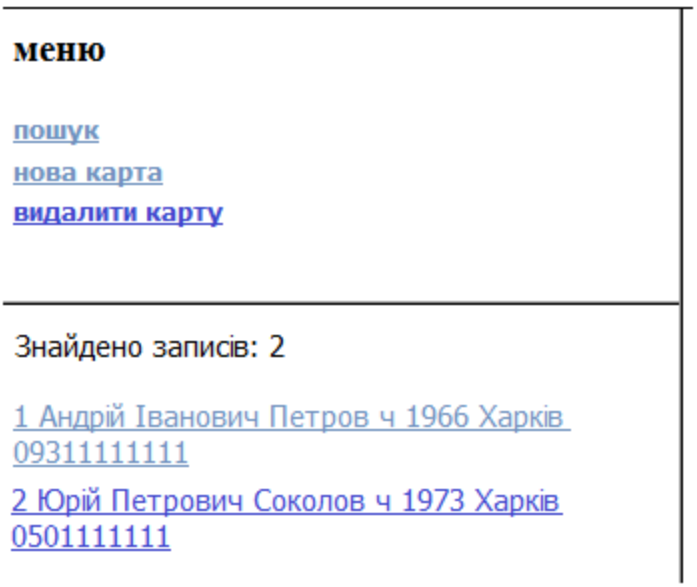

Рис. 3.4. Результат пошуку картки пацієнта

Такий спосіб відображення інформації дозволяє швидко переглядати інформацію для кожного пацієнта, що відкривається в сусідніх фреймах справа. Результат пошуку фіксується у фреймі **patrec**.

3.5.2 Пошук у таблиці '**visit**'

Наступний крок виконує пошук у таблиці **visit**. Після вибору потрібної карти пацієнта та натиснення на ній, керування передається файлу **visit.php**. На початку файлу ми відкриваємо сесію, встановлюємо з'єднання з базою даних, а потім ініціалізуємо змінні, отримані з файлу **search.php** (id, стать, дата народження).

```
<?php
session_start();
     include('connect_db.php');
$idpat=$_GET['idpat'];
$sex=$_GET['sex'];
$year_b=$_GET['year_b'];
```

Ми повинні отримати рік народження з дати, для чого застосовуємо функцію explode(). Вона поділяє рядок за вказаним роздільником. За замовчуванням формат дати на сервері записується за допомогою символу “-“. Ці кроки реалізуються за допомогою коду:

```
$date_elements  = explode("-",$year_b);
$year_b=$date_elements[0];
$_SESSION['year_b']=$year_b; //збереженовсесії
```

Якщо запис знайдено, то id, що передається, не буде порожнім. Ці умови необхідно перевірити, щоб забезпечити обробку винятків та невизначеностей, які можуть призвести до появи помилок або ускладнити

роботу з додатком. Тобто, перехід за гіперпосиланнями можливий лише у випадку, якщо записи існують:

```
if(isset($idpat))
{
//если карта найдена
}
```

Заголовок фрейму **visit**:

```
echo "<h4>Визиты</h4>";
```

потім відбираємо записи, що відповідають обраній карті пацієнта:

```
$sql="SELECT * FROM `visit` WHERE `id_p`='$idpat'";
$rs=mysqli_query($link,$sql);
```

Якщо записи з таблиці відвідувань знайшли **(mys-qli_num_rows()** не дорівнює 0), ми відображаємо їх у вигляді таблиці гіперпосилань. Тут використовуємо той самий прийом, що й у попередньому пункті для файлу **search.php**: передаємо id вибраного запису файлу **patrec.php**, щоб здійснювати пошук записів у таблицях **spiro** та **xray**.

```
if(mysqli_num_rows($rs)>0)
{
echo"<table width='100%'>";
echo"<tr><th                        width='15%'>дата</th><th
width='30%'>лікар</th><th           width='15%'>маса</th><th
width='15%'>зріст</th></tr>";
//друк рядків таблиці відвідувань у вигляді гіперпосилань
while($visit=mysqli_fetch_array($rs))
                    {
```

```
$row="<tr align='center'><td><a
href='patrec.php?idvisit=".$visit['id']."&mass=".$visit['mass'].
"&height=".$visit['height']."&vdate=".$visit['date']."'
id='".$patient['id']." onClick='getid(this)'
target='patrec'>".$visit['date']."</a></td><td>".$visit['doctor'
]."</td><td>".$visit['mass']."</td><td>".$visit['height']."</td>
</tr>";
echo $row;
     }}
     }
     ?>
```

Результат пошуку показано на рис. 3.5.

Пульмонологічна база даних

Вхід в акуант Зареєструватися

меню

пошук
нова карта
видалити карту

Знайдено записів: 1

1 Андрій Іванович Петров ч 1966 Харків
0931111111

Візит

дата	Лікар	маса, кг	зріст, см
2016-01-09	Іванов П.П.	88	182
2014-01-15	Іванов П.П.	89	182

Рис. 3.5. Візити до лікаря

За допомогою функції **mysql_fetch_array ()** результати пошуку поміщені в асоціативний масив **$ visit**. При необхідності будь-який його елемент може бути переданий через сесію в інший файл.

3.5.3. Пошук у таблицях **'spiro'** та **'xray'**

Після вибору візиту до лікаря управління передається файлу **patrec.php** (див. рис. 2.5). Для зручності ми вирішили розмістити дані з таблиць на різних вкладках, що й реалізували за допомогою коду:

```
<html>
<h1>Карта пацієнта</h1>
<head>
<link rel="stylesheet" href="CSS/css.css">
</head>
<div class="tabs">


<input id="tab2" type="radio" name="tabs">
<label for="tab2" title="Spirometry">Спірометрія</label>

<input id="tab3" type="radio" name="tabs">
<label for="tab3" title="Conclusion">Висновок</label>

<input id="tab4" type="radio" name="tabs">
<label for="tab4" title="X-Ray">Рентген</label>

//зміст кожної вкладки
</div>
```

Дизайн вкладок ми виконуємо за допомогою таблиць CSS. Наприклад пояснимо, як саме організована вкладка 2 (tab2, section 2, спірометрія). Повний листинг файлу наводиться у додатку Е.

На початку файлу відкриваємо сесію, щоб отримати **id** з таблиці **visit**, а також інші дані, які будуть необхідні для обчислень. У числі функцій цього сервісу може бути розрахунок спірометричних показників у нормі, враховуючи персональні параметри: вік, зріст, масу тіла. У першому розділі описали їх, включаючи методи обчислень.

```
<?php
session_start();
$idvis=$_GET['idvisit'];
$sex=$_SESSION['sex'];
$year_b=$_SESSION['year_b'];
$mass=$_GET['mass'];
$vdate=$_GET['vdate'];
$height=$_GET['height'];
$date_elements  = explode("-",$vdate);
$vdate=$date_elements[0];
$age=abs($year_b-$vdate);//Розрахуноквікупацієнта
include('connect_db.php');//зв’язок з базою даних
?>
```

Наступний код додає дані із таблиці spiro.

```
<section id="content2">
<a name="section2"></a>
<div>
<?php
echo "<h5>Нормальні значення, вираховані для пацієнта, вказані у дужках</h5>";
$sql="SELECT * FROM `spiro` WHERE `id_v` =$idvis";
$rs=mysqli_query($link,$sql);
```

Якщо запис з таблиці **spiro** був знайдений, ми відобразимо його. У той же час, буде проведено розрахунок нормальних спірометричних показників, описаних у пункті 1.3.1. Змінні мають такі самі імена, але з індексом 'n', тобто норми. Формули розрахунку деяких показників відрізняються для чоловіків і жінок, тому програмно необхідно використовувати розгалуження за допомогою умов **if**.

```
if(mysqli_num_rows($rs)>0)
{
while($row=mysqli_fetch_array($rs))
{
if($sex=='м')
{
  $vcn=(27.63-0.112*$age)*$height;
  $tlcn=(36.2-0.06*$age)*$height;
  }
if($sex=='ж')
{
  $vcn=(21.78-0.101*$age)*$height;
  $tlcn=(28.6-0.06)*$age*$height;

}
$tdn=$mass*10;
$rvn=0.25*$tlcn;
$irvn=0.34*$vcn;
$ervn=0.25*$vcn;
$icn=$IRVn+$tvn;
$rvn=0.25*$tlcn;
$frvn=$rvn+$ervn;

echo "Дихальнийоб’єм (ДО), мл=".$row['TD']."
(".$tdn.")<br>";
echo "Життєва ємність легень (ЖЄЛ), мл=".$row['VC']." (
".$vcn.")<br>";
echo "Загальна ємність легень (ЗЄЛ), мл=".$row['TLC']." (
".$tlcn.")<br>";
echo "Резервний об’єм вдиху(Ровд), мл=".$row['IRV']." (
".$irvn.")<br>";
echo " Резервний об’єм видиху (Ровид), мл=".$row['ERV']." (
".$ervn.")<br>";
```

```
echo "Ємність вдиху (Євд), мл=".$row['IC']." (".$icn.")<br>";
echo "Залишковий об'єм (ЗО), мл=".$row['RV']." (".$rvn.")<br>";
echo "Функціональна залишкова ємність (ФЗЄ), ml=".$row['FRV']." ( ".$frvn.")<br>";
    }
    }
    ?>
    </div>
    </section>
```

Результат роботи коду показано на рис. 3.6.

Рис. 3.6. Результат запиту до таблиці "spiro"

Розглянемо тепер вкладку **tab4** (section 4, ретнгент). Ми використовували такий код, щоб організувати цю вкладку. Запит відправляємо в таблицю **xray**.

```
<section id="content4">
<p>
<?php
```

```
$sql="SELECT * from `xray` where `id_v`=$idvis";
$rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
    {
    $row_xray=mysqli_fetch_array($rs);
    $fimg=$row_xray['img'];
echo $fimg;echo "<br>";
echo "<IMG src='$fimg'>";
echo "<br>";
echo "Описаниеизображения: ".$row_xray['description'];
    }
?>
</p>
</section>
```

Рис. 3.7. Результат запиту в таблиці xray

У наведеному вище коді ми завантажуємо малюнок за допомогою стандартного тега **<IMGsrc='$fimg'>**. Шлях до зображення заноситься в змінну **$fimg**, а самі зображення зберігаються в окремому каталозі на цьому ж сервері. Недоліком є те, що слід виключити випадкове дублювання імен.

Остання вкладка **tab3** (section3, висновок) містить звичайний текст, який витягується з таблиці **visit**:

```
<section id="content3">
<p>
<?php
    $sql="SELECT `conclusion` from `visit` where `id_v`=$idvis";
    $rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
    {
    $row_conc=mysqli_fetch_array($rs);
    echo $row_conc['conclusion'];
    }
?>
</p>
</section>
```

3.6. Створення нової картки пацієнта

Якщо користувач вибирає пункт меню **"нова карта"**, файл **new_pat_form.php** вивантажується у фрейм **patient**. Ми не наводитимемо коду форми, а вкажемо тільки імена полів. Після натискання на кнопку дані з форми передадуть файлу **new_pat.php** для обробки та запису до бази (рис. 3.8).

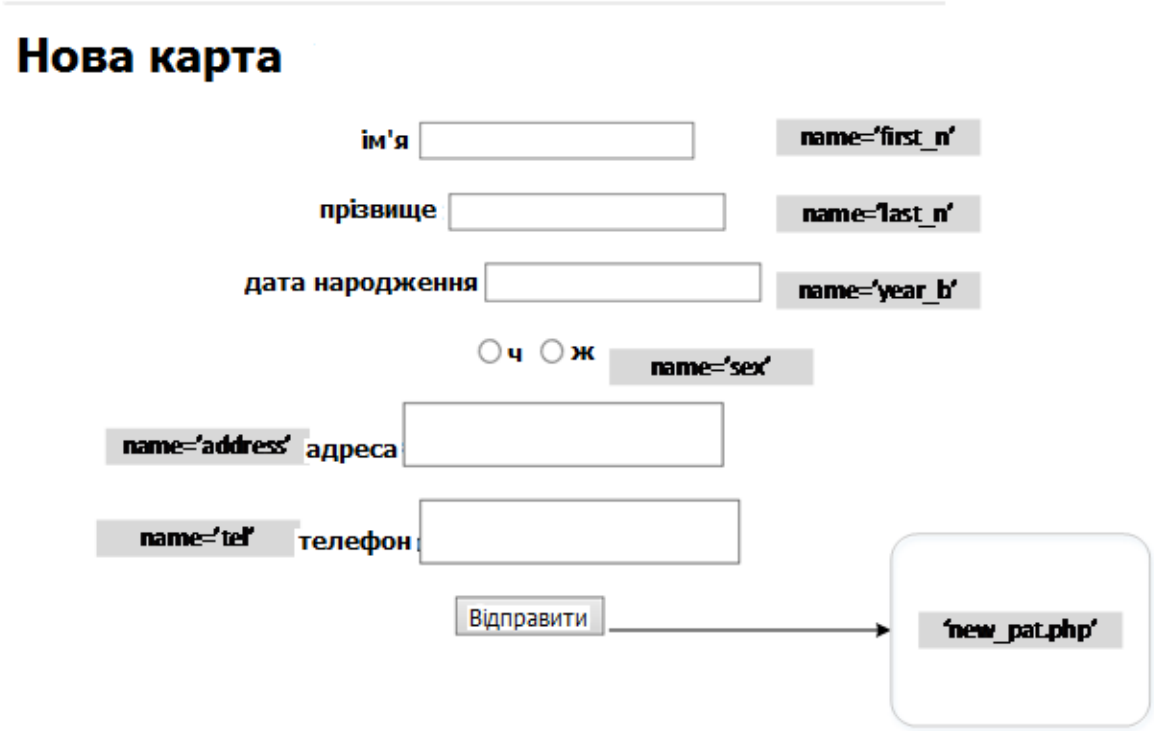

Рис. 3.8. Форма “new_pat_form.php”

Файл **new_pat.php** забирає дані з форми та відправляє їх до бази для запису до таблиці **patient**. Затеон отримує **id** вставлений запис, який необхідний для підпорядкованої таблиці **visit**.

Нижче поданий блок команд вставляє новий запис у таблицю і отримує **id**, який присвоюється змінною **$idpat**.

```
$sql="INSERT INTO `$table`(`first_n`, `last_n`, `year_b`, `sex`,
`address`, `tel`)
VALUES('$first_n','$last_n','$year_b','$sex','$address','$tel')"
;

$rs=mysqli_query($link,$sql);

echo "<script
type='text/javascript'>alert('Записбувдоданий');</script>";

$idpat=mysqli_insert_id($link);//id доданогозапису
```

```
$_SESSION['idpat']=$idpat;

?>
<a href='visit_new.php' target='date'>Додатиіншідані</a>
```

Форма **visit_new.php** містить поля, показані на рис. 3.9.

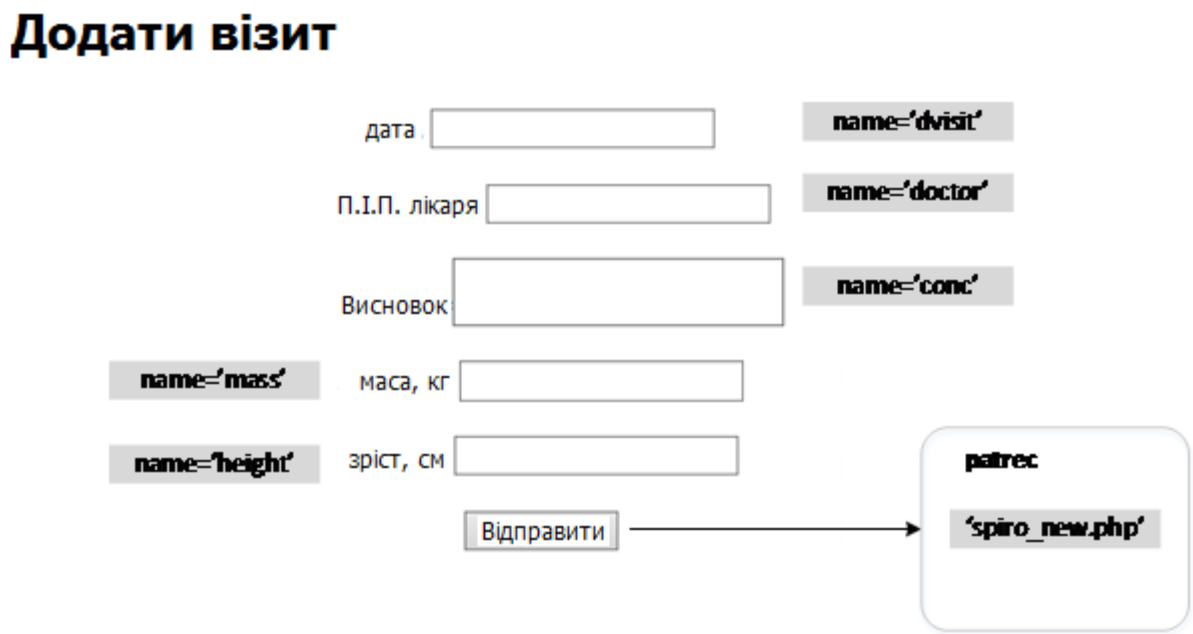

Рис. 3.9. Форма “visit_new.php”

При натисканні на кнопці ми будемо перенаправлені до файлу **spiro_new.php**, який вивантажиться у кадр **patrec**. На початку файлу отримуємо дані з форми, потім встановлюємо зв'язок із базою даних:

```
<?php

session_start();
$idpat=$_SESSION['idpat'];
$dvisit=$_GET['dvisit'];
$doctor=$_GET['doctor'];
$mass=$_GET['mass'];
$height=$_GET['height'];
```

```
$conc=$_GET['conc'];

$link=$_SESSION['link'];
include('connect_db.php');
```

Потім формуємо рядок для SQL-запиту в таблицю **visit**:

```
$sql="INSERT INTO
`visit`(`id_p`,`date`,`doctor`,`mass`,`height`,`conclusion`)
VALUES('$idpat','$dvisit','$doctor','$mass','$height','$conc')";
$rs=mysqli_query($link,$sql);

$idvisit=mysqli_insert_id($link);
$_SESSION['idvis']=$idvisit;

?>
```

Наступний крок – введення результатів спірометричних досліджень та збереження рентгенівських зображень. Ми вирішили помістити їх на одній формі, вид якої показано на рис. 3.10.

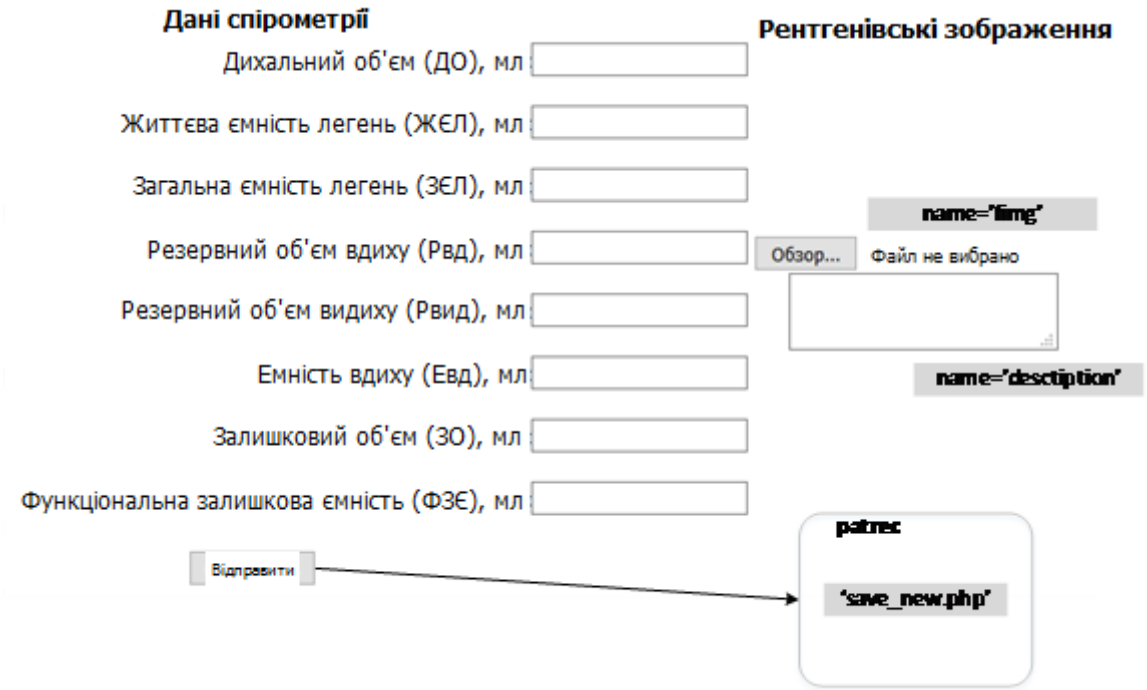

Рис. 3.10. Форма “**spiro_new.php**”

Дані у формі розміщуються у дві колонки та мають табличне представлення. Форму реалізовано способом, аналогічним розглянутим вище. Імена полів для спірометричних даних збігаються з іменами полів таблиці **spiro** (див. рис. 2.3).

Під час вибору рентгенівського зображення відкривається стандартне діалогове вікно. Ми скористалися новою функцією з арсеналу HTML5:

```
<inputtype='file' name='fimg' value'Виберітьзображення>
```

Розглянемо файл **save_new.php**, якому передається управління після натискання на кнопку. Спочатку ми відкриваємо сесію, а потім отримуємо дані з форми **spiro_new.php** за протоколом GET:

```
<?php
session_start();
echo "idvisit=".$idvis;
$td=$_GET['td'];
$vc=$_GET['vc'];
$tlc=$_GET['tlc'];
$irv=$_GET['irv'];
$erv=$_GET['erv'];
$ic=$_GET['ic'];
$rv=$_GET['rv'];
$frc=$_GET['frc'];
$fimg=$_GET['fimg'];
$descr=$_GET['descr'];
include('connect_db.php');//з’єднання з базою даних
```

Далі створюємо рядки запиту для додавання даних до таблиць **spiro ($sql)** та **xray ($sql1)**

```
$sql="INSERT into
`spiro`(`id_v`,`td`,`vc`,`tlc`,`irv`,`erv`,`ic`,`rv`,`frc`)
```

```
VALUES('$idvis','$td','$vc','$tlc','$irv','$erv','$ic','$rv','$f
rc')";
$rs=mysqli_query($link,$sql);

$sql1="INSERT into `xray`(`id_v`,`img``description`)
VALUES('$idvis','$fimg','$descr')";
$rs1=mysqli_query($link,$sql1);
?>
```

3.7. Видалення картки пацієнта

Видалення картки вимагає наступних кроків:

1) пошук картки пацієнта (форма **“search_del.php”**);

2) вибір карти зі списку знайдених записів (файл **"del_pat.php"**);

3) виконання запиту видалення (файл **“remove_query.php”**).

Файл **search_del.php** має ту саму структуру, що і файл **search_form.php**. Різниця полягає в тому, що після натискання на кнопку відкривається файл **del_pat.php**. Останній знаходить записи у таблиці **patient**, формує список рядків (рис. 3.11).

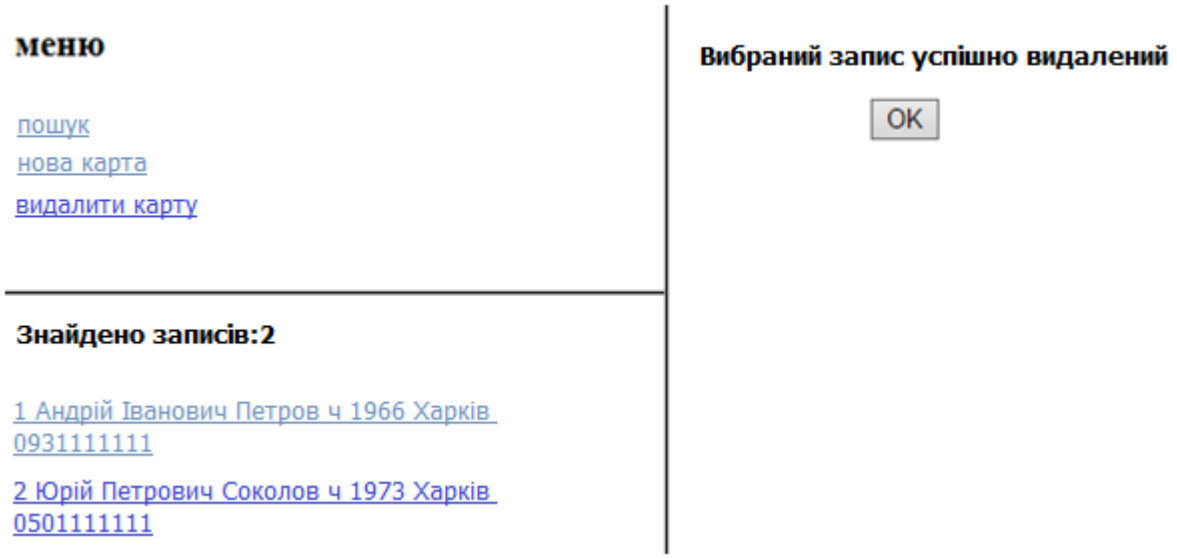

Рис. 3.11. Видалення картки пацієнта

При натисканні на записи (гіперпосилання) відбувається перенаправлення на “remove_query.php”, що виконує запит на видалення поточного запису. Лістинг файлу наводиться у додатку Е.

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

В атестаційній роботі розглянуто питання створення інформаційної системи на кшталт «клієнт-сервер» для допомоги лікарю-пульмонологу. Основне призначення ПП – мережне зберігання даних та медичний калькулятор показників.

4.1. Опис програмного продукту

Комплекс програмного забезпечення має інтерфейс браузерного типу, базу даних, розміщену на зовнішньому сервері та систему управління базою даною, реалізованою мовами HTML 5.0, php 7.1.

Програмний продукт (ПП) орієнтований користувачів, мають мінімальний досвід роботи з персональної технікою (комп'ютери, планшети).

Розроблена система проста у використанні, доступ до даних може здійснювати за допомогою глобальної мережі.

ПП висуває такі системні вимоги:

- ПК типу IBM PC з урахуванням процесорів Intel чи сумісних з ними;
- об'єм жорсткого диска не менше 300 Мб;
- мінімальний обсяг ОЗУ 32 Мб;
- операційна система Microsoft Windows XP та вище;
- відеоадаптер SVGA або монітор з більш високою роздільною здатністю;
- підтримка браузера: Internet Explorer, Mozilla Firefox та Opera.

Використання комп'ютерів з великими обчислювальними можливостями підвищує зручність роботи з програмним комплексом.

ПП орієнтований на такі категорії споживачів: середній медичний персонал, лікарі.

4.2. Оцінка ринку збуту

За оцінками Frost&Sullivan, у 2010 р. загальний обсяг витрат на IT в індустрії охорони здоров'я становив 310 млн. дол. США. Прогнозується, що за декілька десятків років він досягне 882 млн. дол. США та більше при сукупньому середньорічному темпі зростання (CAGR) у 16,1%.

Результат розрахунку ємності ринку ПП по Харківській області представлено у табл. 4.1.

Таблиця 4.1.

Аналіз ємності ринку ПП

Галузь застосування	**Кількість об’єктів**	**Очікувана кількість екземплярів одному користувачеві**	**Передбачувана ємність ринку**
Лікувальні заклади	22	2	44
Діагностичні центри	23	2	46
Всього	45	45	90

Отже, ринок цього продукту складається з двох сегментів.

4.3. Розрахунок витрат за розробку ПП

Для розрахунку ціни товару необхідно визначити витрати на розробку програмного засобу. Ці витрати обчислюємо, розрахувавши трудомісткість виконання робіт (табл. 4.2). Вся технологія була розбита на 9 основних етапів. Кожен етап був визначений за часом та сумою заробітної плати (ЗП) розробників ПП [11, 12].

Таблиця 4.2.

Розрахунок трудомісткості розробки ПП та ЗП розробника

Вид роботи	**Виконавець**		**Праце-витрати, люд.-день**	**Середньо-денна ЗП, грн./люд.-день**	**Сума ЗП, грн. (гр.3хгр.4хгр.5)**
	Посада	**Кількість людей**			
1. Розробка технічного завдання	Керівник проєкту	1	3	418,18	1254,54
2. Підбір та вивчення літературних джерел	Інженер-програміст	1	5	231,82	1159,1
3. Вибір і узгодження ПЗ	Керівник проєкту	1	1	418,18	418,18
	Інженер-програміст	1	1	231,82	231,82
4. Розробка інтерфейсу користувача	Інженер-програміст	1	2	231,82	463,64
5. Розробка бази даних и СУБД	Інженер-програміст	1	10	231,82	2318,2
6. Розробка модулю калькулятору показників	Інженер-програміст	1	3	231,82	695,46
7. Тестування ПЗ	Керівник проєкту	1	2	231,82	463,64
8. Оформлення технічної документації	Інженер-програміст	1	5	231,82	1159,1
Разом (ЗП)			32		8163,68

Середня заробітна плата виконавця ($З_{СД}$) розраховується за формулою:

$$З_{СД} = \frac{З_{мес}}{n}, \qquad (4.1)$$

де $З_{міс}$ – місячна заробітна плата виконавця, грн.;

n – кількість робочих днів на місяць (n = 22 дні).

Знаходимо середньоденну зарплату інженер-програміста:

$$З_{СД} = \frac{5100}{22} = 231{,}82 \text{ грн.}$$

Знаходимо середньоденну зарплату керівника:

$$З_{СД} = \frac{9200}{22} = 418{,}18 \text{ грн.}$$

Єдиний соціальний внесок (ЕСВ) на загальнообов'язкове державне соціальне страхування – це консолідований за всіма видами загальнообов'язкового державного соціального страхування страховий внесок.

Відрахування на соціальні заходи до відповідних фондів здійснюються чинним законодавством України. Розрахуємо ЄСВ, що становить 22% від ЗП. Отже, ЄСВ становить 1079,01 грн.

Матеріальні витрати – це витрати на придбання малоцінних та швидкозношуваних предметів, які включають вартість лазерних дисків, картриджа, канцелярських виробів, Flash-пам'яті, паперу для друку, олівців тощо.

Таблиця 4.3.

Розрахунок матеріальних витрат за розробку ПП

Назва продукту	**Кількість використаних одиниць продукту**	**Ціна за 1-цю продукту, грн**	**Загальні витрати, грн**
Ручка кулькова, шт.	1	2,24	2,24
Степлер, шт.	1	39,14	39,14
Папка з затискачем	1	21,85	21,85
Flash-накопичувач, 8 Гб	1	117	117
Офісний папір А4, упаковка (500 листів)	1	64,78	64,78
Тонер для принтера, шт.	1	120	120
Разом:			365,01

Вартість машинного часу ($В_{мч}$) розраховується за формулою:

$$В_{Мч}=t_{Мч}\cdot З_{Мч}, \quad (4.2)$$

- де $t_{Мч}$ – вартість однієї години роботи на ПЕОМ, грн.;
- $З_{Мч}$ – загальний час роботи на ПЕОМ, год.

Оскільки вартість однієї години роботи на ПК становить 0,4667 грн., а загальний час роботи на ПК становив 256 годин (32 чол.-день із табл. 4.2), тоді розраховуємо амортизацію основних фондів:

$$З_{Мч} =0,4667\cdot 256=119,48 \text{ грн}$$

Загальногосподарські витрати включають плату за комунальні послуги: водопостачання, водовідведення, підігрів води, електроенергія та складають 25% від основної заробітної плати.

$$З_{заг.госп.}=0,25\cdot З_{заг.}, \quad (4.3)$$

де $З_{заг.госп.}$ – загальна зарабітня плата, грн.

$$З_{заг.госп.} = 0,25\cdot 8163,68 = 2040,92 \text{ грн.}$$

Вартість оплати послуг зв'язку, що надаються інтернет-провайдерами – 120 грн., операторами телефонного зв'язку – 45 грн. Отже, вартість оплати послуг зв'язку становить 165 грн.

Витрати на реалізацію маркетингових заходів (наприклад, проведення маркетингових заходів оцінки рівня конкурентоспроможності ПП, розрахунку територіальної ємності (обсягу) ринку, просування нового ПП)

становлять 15% від основної заробітної плати. Розрахунок витрат за розробку ПП наведено у табл. 4.4.

$$З_{маркет.} = 0{,}15 \cdot З_{осн.} = 0{,}15 \cdot 8163{,}68 = 1224{,}55 \text{ грн.}$$

Таблиця 4.4.

Розрахунок витрат за розробку ПП

№ з/п	Стаття витрат	Значення, грн.
1.	Заробітна плата	8163,68
2.	Єдиний соціальний внесок (22 % від ФОТ)	1796,01
3.	Матеріальні витрати	218,58
4.	Амортизація основних фондів (вартість машинного часу)	119,48
5.	Інші витрати, в тому числі:	
5.1	загальногосподарські витрати	2040,92
5.2	вартість послуг зв’язку	165
6.	Витрати на маркетингові заходи	1224,55
7.	Разом ($З_р$)	13728,22
8.	Витрати на розробку одиниці ПП	152,94

Розрахунок витрат за розробку одиниці ПП (З) визначається за формулою:

$$З = \frac{З_р}{М_р}, \tag{4.4}$$

де $З_р$ – загальні витрати на розробку ПП, грн.;

$М_Р$ – ємність територіального ринку ПП, шт.

$$З = \frac{13728{,}22}{90} = 152{,}94 \text{ грн.}$$

Розрахунок відпускної ціни однієї копії ПП наведено у табл.4.5.

Таблиця 4.5.

Розрахунок витрат на тиражування та відпускної ціни однієї копії ПП

№ з/п	Стаття витрат	Значення, грн.
1.	Розмір заробітної плати з нарахуваннями	1,7
2.	Витрати на просування одиниці ПП	14,44
3.	Витрати на адаптацію ПП (25%)	0,42
4.	Витрати на розробку одиниці ПП (п. 8 таб. 5.4)	152,94
5.	Собівартість однієї копії (п. 4 + п. 3)	152,96
6.	Прибуток запланований (30 % від п. 5)	45,89
7.	Розмір НДС (20% от (п. 6 + п. 7))	39,77
8.	Відпускна ціна однієї копії ПП (п. 5 + п. 6 + п. 7)	408,12

Розрахунок витрат на просування ПП наведено в табл.4.6.

Таблиця 4.6.

Витрати на просування ПП

Рекламний захід	**Вартість, грн.**
1. Програма знижок	500
2. Пряма реклама	300
3. Банерна реклама в Інтернеті	300
4. Інші	200
Разом	1300
Витрати на просування одиниці ПП	14,44

Витрати на просування одиниці ПП ($З_{пр}$) розраховується за формулою:

$$З = \frac{З_{пр}}{О_{тер}}, \qquad (4.5)$$

де

- $З_{пр}$ – сумарні витрати на просування ПП, грн;
- $О_{тер}$ – об’єм територіального ринку, шт.

$$З = \frac{1300}{90} = 14{,}44 \text{ грн.}$$

4.4. Оцінка рівня конкурентоспроможності ПП

Конкурентоспроможність програмного продукту визначається не його «абсолютною цінністю» для споживача, а порівняльною корисністю його характеристик (технічних та економічних параметрів) із відповідними характеристиками ПП конкурента.

Технічні та економічні показники ПП наведені у таб.4.7.

Таблиця 4.7.

Технічні та економічні параметри ПП

Параметр	**Новий ПП**	**ПП конкурента**
Об’єм необхідного дискового простору для роботи ПП	500 Мбайт	1000 Мбайт
Складність програмного продукту:		
- складність інтерфейсу користувача (число активних вікон)	1	3
- виконувані функції	15	12
- об’єм використаних даних (бази даних, файли даних, довідники, словники)	1 база даних	1 база даних
- об’єм виконуваного коду	48 Мбайт	56 Мбайт
Доступність використання:		
- по простоті запуску	3 бали	3 бали
- по мультимедійному забезпеченню	3 бали	3 бали
Супроводжуюча документація	2 бали	2 бали
Засоби навчання	1 розділ	2 розділи
Довідкова система	1 розділ	1 розділ
Ціна програмного продукту	408,12 грн.	550 грн.

Показник рівня конкурентоспроможності нового ПП (КСП) розраховується за формулою:

$$КСП = \frac{I_T}{I_Э} > 1, \tag{4.6}$$

де,

IT – зважений параметричний показник за технічними параметрами;

IE – зважений параметричний показник за економічними параметрами.

Результат розрахунку конкурентоспроможності ПП наведено у таб.4.8.

Таблиця 4.8.

Розрахунок конкурентоспроможності ПП

Параметр	**Індекс покращення параметра***	**Значення параметра**		**Ранг параметра**	**Одиничний параметричний індекс (гр.3/гр.4)*, (гр.4/гр.3)****	**Зведений параметричний показник (гр.5хгр.6)**
		продукція нового ПП	**Продукція конкурента**			
1	2	3	4	5	6	7
1. Технічні параметри						
1.1 Розмір програмного коду, Мб	**	48	56	0,20	0,85	0,17
1.2. Об’єм оперативної пам’яті ПК, Мб	**	1012	1012	0,30	1	0,3
1.3. Функціональність	*	15	12	0,5	1,25	0,625
Разом:				1		1,1
2. Економічні параметри						
2.1. Ціна ПП, грн.	*	408,12	550	0,5	0,74	0,37
2.2. Затрати на навчання персоналу, грн.	**	200,00	200	0,3	1,00	0,30
2.3. Сервісне обслуговування ПП, грн.	**	200,00	250	0,2	0,80	0,16
Разом:				1		0,83

Якщо збільшення параметра веде для його поліпшення, то індекс дорівнює *, якщо погіршення, то індекс дорівнює **.

$$КСП = \frac{1,1}{0,83} = 1,32.$$

Коефіцієнт конкурентоспроможності товару (КСП) дорівнює 1,34. Так як КСП>1, то рівень конкурентоспроможності нового ПП більше по відношенню до ПП конкурента.

4.5. Аналіз ризику ПП

При аналізі ризику виведення ринку програмного продукту доцільно розрахувати показник, що дозволяє визначити, наскільки розробник може знизити обсяг продажу, не зазнавши збитків. Він визначається як відношення різниці між очікуваним обсягом продажу та точкою беззбитковості до очікуваного обсягу продажу.

Точка беззбитковості (ТБЗ) – це обсяг продажів, у якому валова виручка (дохід) від реалізації покриває загальні сукупні постійні витрати і визначається за формулою:

$$Т_{БЗ} = \frac{З_{пост}}{Ц - З_п}, \qquad (4.7)$$

де,

$З_{пост}$ – постійні витрати на розробку та реалізацію ПП, грн.;

Ц – розрахункова ціна ПП, грн.;

$З_п$ – змінні витрати на одиницю ПП, грн.

$$T_{БЗ} = \frac{13728{,}22}{408{,}12 - 14{,}44} = 35 \text{ шт.}$$

Показник ризику розробника ПП розраховується за такою формулою:

$$П_р = \frac{Ц \cdot M_р - T_{бз}}{Ц \cdot M_р} \cdot 100\%, \tag{4.8}$$

де,

$M_Р$ – очікуваний обсяг продажів (ємність ринку), шт.;

Ц – розрахункова ціна ПП, грн.;

$Т_{БЗ}$ – точка беззбитковості (шт).

$$П_р = \frac{90 - 35}{90} \cdot 100\% = 61\%.$$

Чим вище є чисельне значення показника $П_р$, тим менш ризикованою є робота розробника ПП.

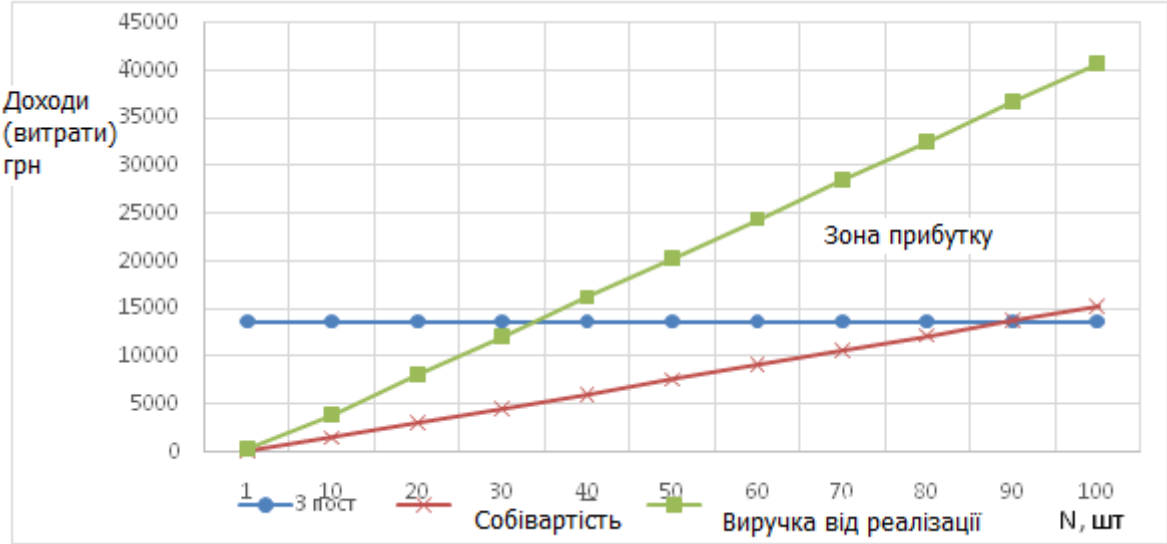

Рис. 4.1. Графік беззбитковості

Зобразимо графік беззбитковості для варіанта, коли ціна на виріб становить 408,12 грн., а умовний обсяг реалізації – 35 шт., тоді пряма 1 – постійні витрати, що становлять 13728,22 грн. собівартості виробу (рис.4.1).

ВИСНОВКИ

1. Проведено медико-технічне обґрунтування роботи. Розглянуто основні питання, що описують фізіологію дихання, інструментальні методи дослідження легень. Це дозволило вибрати список показників, який необхідно включити до карти пацієнта, а саме: результати спірометричних та рентгенівських досліджень.
2. Розроблено базу даних, структура логічно випливає з розглянутого в першому розділі матеріалу. Тип бази даних – реляційний, кількість таблиць – 4.
3. Розроблено структуру інтерфейсу браузерного типу, вибір якої пояснюється цільовим застосуванням модуля – мережеве зберігання даних.
4. Розроблено алгоритмічне та програмне забезпечення модуля, який побудовано за технологією «клієнт-сервер». Одна з функцій модуля – розрахунок нормальних значень легеневих ємностей та обсягів з урахуванням персональних даних пацієнта: зростання, вік, маса тіла та стать. Це полегшує інтерпретацію даних лікарем.

   Програмна реалізація модуля виконана мовою php 7.0*, для відображення форм та статичної інформації на сторінках використовувався HTML 5.0. Доступ та маніпуляція даними здійснювалися за допомогою SQL. База даних та модуль розміщені на одному веб-ресурсі та доступні за адресою: https://diplomaproject.000webhostapp.com/index.php
5. Проведено техніко-економічне обґрунтування роботи, обчислено ціну програмного продукту, яка становить 408,12 грн. Визначена точка беззбитковості виробництва, яка дорівнює 35 копій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авраменко В.І. Формування основних напрямків розвитку інформаційних технологій в охороні здоров'я України на основі світових тенденцій / В.І. Авраменко, В.О. Качмар // Український журнал телемедицини та медичної телематики. – 2011. – Т.9, №2. – С. 124-133.

2. Азархов О.Ю. Процедура отримання діагностичної інформації для медичних інформаційних систем / О. Ю. Азархов, А. П. Моторний, Д. Х. Штофель // Український журнал телемедицини та медичної телематики. – 2011. – Т. 9, № 2. – С. 161-165.

3. Глушко Л.В., Федоров С.В., Скрипник І.М. Внутрішня медицина: у 2 частинах. Частина 1. Розділи 1-8: підручник. Київ, 2019. 680 с.

4. Григус І. М. Фізична реабілітація в пульмонології: навчальний посібник. – Вид. 2, виправлене. – Рівне: НУВГП, 2018. - 258 с.

5. Здоров'я 2020: український вимір : Концепція Загальнодержавної програми від 31 жовтня 2011 р. № 1164-р. [Електронний ресурс]. – Режим доступу : http://zakon3.rada.gov.ua/laws/show/1164-2011-%D1%80.

6. Ілащук Т.О., Бачук-Понич Н.В., Васюк В.Л., Мікулець Л.В. Методи дослідження органів дихання. – Чернівці. -2018. -192 с.

7. Інформаційні технології у сфері охорони здоров'я : монографія / Л.Б. Ліщинська, С.А. Яремко, К.В. Копняк, І.О. Гулівата, Л.П. Гусак ; за заг. ред. Л.Б. Ліщинської. – Вінниця : видавничо редакційний відділ ВТЕІ КНТЕУ, 2018. – 240 с.

8. Качмар В.О. Електронна медична карта пацієнта. Взаємосумісність та стандартизація / В.О. Качмар, А.І. Хвищун // Український журнал телемедицини та медичної телематики. – 2008. – Т. 6, №1. – С. 76-79.

9. Качмар В.О. Медичні інформаційні системи – стан розвитку в Україні / В.О. Качмар // Український журнал телемедицини та медичної телематики. – 2010. – Т. 8, №1. – С.12-17.

10. Кисельов С. М., Сиволап В. Д., Земляний Я. В. Діагностика та лікування захворювань органів дихання у людей похилого віку. Запоріжжя: ЗДМУ, 2020. – 83 с.

11. Методичні вказівки до виконання економічної частини дипломних робіт для студентів усіх форм навчання спеціальності ПЗАС/ упоряд.: Н.Б.Івченко, А.В.Суботіна. – Харків: ХНУРЕ, 2004. – 24 с.

12. Методичні вказівки до виконання економічної частини дипломних робіт для студентів усіх форм навчання спеціальностей: 7.091001 – Виробництво електронних засобів; 7.091002 – Біотехнічні та медичні апарати і системи; 7.091003 - Побутова електронна апаратура; 7.091004 – Технології та засоби телекомунікацій/ упоряд.: В.Г. Харченко, О.І. Горбач, Т.В. Божко – Харків: ХНУРЕ, 2004. – 20 с.

13. Мінцер, О.П. Оброблення клінічних і експериментальних даних у медицині/ О.П. Мінцер. – К.: Вища шк., 2003. – 350 с.

14. Основи діагностики лікування та профілактики основних хвороб органів дихання : навчальний посібник для студентів в галузі знань 22 «Охорона здоров'я» спеціальностей 222 «Медицина», 228 «Педіатрія» навчальна дисципліна «Внутрішня медицина» / С. М. Кисельов, О. В. Назаренко, Я. В. Земляний, Н. І. Капшитар, Є. О. Аравіцький. – Запоріжжя : ЗДМУ, 2021. – 153 с.

15. Основні клінічні синдроми в пульмонології : навчальний посібник для лікарів-інтернів за спец. «Загальна практика-сімейна медицина», «Внутрішні хвороби» / В. І. Кривенко, І. В. Непрядкіна, О. П. Федорова [та ін.]. - Запоріжжя: [ЗДМУ], 2018. - 132 с.

16. Про захист інформації в інформаційно-телекомунікаційних системах : Закон України від 05.07.1994 № 80/ 94-ВР [Електронний ресурс]. - Режим доступу : http://zakon2.rada.gov.ua/laws/show/80/94- %D0%B2%D1%80.

17. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI [Електронний ресурс]. - Режим доступу : http://zakon2.rada.gov.ua/laws/show/2297-17.

18. Сабадаш В. Є. Захворювання дихальної системи. Вибрані питання діагностики та лікування для сімейного лікаря : навчально-методичний посібник; Нац. мед. ун-т ім. О.О. Богомольця МОЗ України. - Київ : Медікс, 2016. - 119 с.

19. Свінціцький А. С. Методи діагностики в клініці внутрішньої медицини : навч. посіб. для лікарів-інтернів і студентів мед. закл. вищ. освіти / А. С. Свінціцький. Київ : Медицина, 2019. - 1007 с.

20. Сиволап В. Д., Земляний Я. В. Інструментальні методи функціональної діагностики захворювань органів дихання. Запоріжжя, 2017. 120 с.

21. Суэринг, С. PHP і MySQL. Біблія программіста: пер. с англ. – Діалектика, 2010. – 912 с.

22. Фещенко Ю. І., Гаврисюк В. К., Дзюблик О. Я. [та ін.] Адаптована клінічна настанова: Хронічне обструктивне захворювання легень. Укр. пульмонол. журнал. 2020, № 3. С. 5-36.

23. GOLD. Global strategy for diagnosis, management and prevention. 2020. [Електронний ресурс]. https://goldcopd.org/wpcontent/uploads/2019/11/GOLD-2020-REPORT-ver1.0wms.pdf.

24. Lerdorf R. Programming PHP / R. Lerdorf, K. Tatroe, P. MacIntyre. – Toronto: O'Reilly Media, 2006. – 542 с.

25. NICE guideline. Chronic obstructive pulmonary disease in over 16s: diagnosis and management. 2019. [Електронний ресурс]. https://www.nice.org.uk/guidance/ng115 6.

ДОДАТОК А

База даних. Схема структурна

ДОДАТОК Б

Система управління базою даних. Схема даних

ДОДАТОК В

Модуль збереження даних.Схема структурована

ДОДАТОК Г

Пошук карти пацієнта. Схема програми

ДОДАТОК Д

Програмні коди

ДОДАТОК Е

Медико-технічне обґрунтування

ДОДАТОК Ж

Техніко-економічне обґрунтування

List-codefor “search_form.php”

```
<h4>Search Patient</h4>
<form action='search.php' method='GET'>
<div align='center'>
first name <br><input name='first_n' type='text'>
<br>



<br>
last name <br><input name='last_n' type='text'/>

<br>
<br>
<input type='radio' name='sex' value='male'>male
<input type='radio' name='sex' value='female'>female<br>
<br>
year_b<br><input type='text' name='year_b'>
<br>
<br>
<div align='center'><input type='submit' value='OK'>
</div>
</form>
```

List-code for “search_.php”

```
<?php
session_start();

include('connect_db.php');
$table='patient';

//Get from the form

$first_n=$_GET['first_n'];


$last_n=$_GET['last_n'];

$year_b=$_GET['year_b'];

$sex=$_GET['sex'];

if(!empty($first_n))
     {
        $flag_FN=1;

     }

if(!empty($last_n))
     {
        $flag_LN=2;
     }

if(!empty($year_b))
     {
        $flag_YB=4;
     }
```

```
     if(!empty($sex))
          {
             $flag_S=8;
          }
     $flags=$flag_FN+$flag_LN+$flag_YB+$flag_S;

         switch ($flags) {
              case 0:
                echo "Fill the fields";
               break;
              case 1:
$sql="SELECT * FROM `$table` WHERE `first_n` like'$first_n%'";
               break;

     //Search by last_n
              case 2:
$sql="SELECT * FROM `$table` WHERE `last_n` like'$last_n%'";
               break;

     //Search by sex
              case 8:
$sql="SELECT * FROM `$table` WHERE `sex`='$sex'";
               break;

     //Search by year_b
              case 4:
$sql="SELECT * FROM `$table` WHERE `year_b` like '$year_b%'";
               break;

     //Search by fist_n and last_n
              case 3:
$sql="SELECT * FROM `$table` WHERE `first_n` like'$fisrt_n%' AND
`last_n` like '$last_n%'";
               break;
```

```
     //Search by last_n and sex
              case 10:
$sql="SELECT * FROM `$table` WHERE `last_n` like '$last_n%' AND
`sex` = '$sex'";
               break;

     //Search by fist_n and sex
              case 9:

$sql="SELECT * FROM `$table` WHERE `first_n` like '$fisrt_n%'
AND `sex` = '$sex'";
               break;

     //Search by first_n and year_b
              case 5:
$sql="SELECT * FROM `$table` WHERE `first_n` like '$first_n%'
AND `year_b` like '$year_b%'";
               break;


     //Search by last_n and year_b
              case 6:

$sql="SELECT * FROM `$table` WHERE `last_n` like '$last_n%' AND
`year_b` like '$year_b%'";
               break;

     //Search by first_n, last_n and year_b
              case 11:

$sql="SELECT * FROM `$table` WHERE `first_n` like '$first_n%'
AND `last_n` like '$last_n%'  AND `year_b` like '$year_b%'";
               break;
```

```
     //Search by last_n, sex and year_b
              case 14:

$sql="SELECT * FROM `$table` WHERE `last_n` like '$last_n%' AND
`sex` = '$sex' AND `year_b` like '$year_b%'";
               break;

     //all
              case 15:

                  $sql="SELECT * FROM `$table` WHERE `first_n`
like '$first_n%' AND `last_n` like '$last_n%' AND `sex` = '$sex'
AND `year_b` like '$year_b%'";
               break;

        }
     $rs=mysqli_query($link,$sql);

     if(!$rs)
     {
     die('Wrong query. Try again: ' . mysql_error());
     }
$num_pat=mysqli_num_rows($rs);
$patient_found=file_get_contents('patient.php');
$patient_found=str_replace("{N}",$num_pat,$patient_found);
echo $patient_found;

while($patient=mysqli_fetch_array($rs))
     {
$row=$patient['id']."  ".$patient['first_n']."
".$patient['last_n']."  ".$patient['sex']."
".$patient['year_b']."  ".$patient['address']."
".$patient['tel'];

echo "<a
href='visit.php?idpat=".$patient['id']."&sex=".$patient['sex']."
```

```
&year_b=".$patient['year_b']."' id='".$patient['id']."'
target='date' onClick='getid(this)'>".$row."</a><br>";
     ?>
```

List-code for “visit.php”

```
<?php
session_start();
include('connect_db.php');
$idpat=$_GET['idpat'];
$sex=$_GET['sex'];
$year_b=$_GET['year_b'];
$date_elements  = explode("-",$year_b);
$year_b=$date_elements[0];
$_SESSION['year_b']=$year_b;

if(isset($idpat)){
$link=$_SESSION['link'];

//TAB DATE OF VISIT
?>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.
min.js"></script>

<script type="text/javascript">



function getid(obj) {
          var e = obj.id;

     $.get("spiro.php", {idvis:e}, function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
}
</script>
```

```
<?php
echo "<h4>Visit</h4>";
?>
<head>
<link rel="stylesheet" href="CSS/css.css">
</head>
<?php
            $sql="SELECT * FROM `visit` WHERE `id_p`='$idpat'";
            $rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
{
echo"<table width='100%'>";
echo"<tr><th
width='15%'>date</th><th>doctor</th><th>mass</th><th>height</th>
</tr>";

       while($visit=mysqli_fetch_array($rs))
                   {

$row="<tr align='center'><td><a
href='patrec.php?idvisit=".$visit['id']."&mass=".$visit['mass'].
"&height=".$visit['height']."&vdate=".$visit['date']."'
id='".$patient['id']." onClick='getid(this)'
target='patrec'>".$visit['date']."</a></td><td>".$visit['doctor'
]."</td><td>".$visit['mass']."</td><td>".$visit['height']."</td>
</tr>";

$_SESSION['sex']=$sex;

echo $row;
                   }

}
}

?>
```

List-code for “patrec.php”

```
<?php
session_start();
$idvis=$_GET['idvisit'];
$sex=$_SESSION['sex'];
$year_b=$_SESSION['year_b'];

$mass=$_GET['mass'];
$vdate=$_GET['vdate'];
$height=$_GET['height'];

$date_elements  = explode("-",$vdate);
$vdate=$date_elements[0];
//echo "year=     ".$vdate;

$age=abs($year_b-$vdate);
//echo "id visit=".$idvis."    sex=".$sex."      mass=".$mass."
height=".$height."       age=".$age;
         include('connect_db.php');

?>

<html>
<h1>Patients record</h1>
<head>
<link rel="stylesheet" href="CSS/css.css">
</head>

<div class="tabs">



<input id="tab2" type="radio" name="tabs">
<label for="tab2" title="Spirometry">Spirometry</label>
```

```
<input id="tab3" type="radio" name="tabs">
<label for="tab3" title="Conclusion">Conclusion</label>

<input id="tab4" type="radio" name="tabs">
<label for="tab4" title="X-Ray">X-ray imaging</label>


<section id="content2">
<a name="section2"></a>
<div>
<?php
echo "<h5>Nornal values, calculated for the person, is given in
parses</h5>";
$sql="SELECT * FROM `spiro` WHERE `num` =$idvis";
$rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
{
while($row=mysqli_fetch_array($rs))
{
if($sex=='male')
{
  $vcn=(27.63-0.112*$age)*$height;
  $tlcn=(36.2-0.06*$age)*$height;

}
if($sex=='female')
{
  $vcn=(21.78-0.101*$age)*$height;
  $tlcn=(28.6-0.06)*$age*$height;

}
$tdn=$mass*10;
  $rvn=0.25*$tlcn;
  $irvn=0.34*$vcn;
  $ervn=0.25*$vcn;
```

```
  $icn=$irvn+$tdn;
  $rvn=0.25*$tlcn;
$frvn=$rvn+$ervn;

echo "Tidal Volume (TD), ml=".$row['TD']."     (".$tdn.")<br>";

echo "Vital Capacity (VC), ml=".$row['VC']."     (
".$vcn.")<br>";
echo "Total Lung Capacity (TLC), ml=".$row['TLC']."     (
".$tlcn.")<br>";
echo "Inspiratory Reserve Volume  (IRV), ml=".$row['IRV']."
( ".$irvn.")<br>";
echo "Expiratory Reserve Volume  (ERV), ml=".$row['ERV']."     (
".$ervn.")<br>";
echo "Inspiratory Capacity   (IC), ml=".$row['IC']."
(".$icn.")<br>";
echo "Residual Volume    (RV), ml=".$row['RV']."     (
".$rvn.")<br>";

echo "Functional residual capacity   (FRV), ml=".$row['FRV']."
( ".$frvn.")<br>";
}
}
?>
</div>
</section>
<section id="content3">
<p>
<?php
$sql="SELECT `conclusion` from `visit` where `id_p`=$idvis";
    $rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
    {
    $row_conc=mysqli_fetch_array($rs);
```

```
    echo $row_conc['conclusion'];
    }
?>
</p>
</section>
<section id="content4">
<p>
<?php
//header('Content-type: image/jpeg');
 $sql="SELECT * from `xray` where `id_v`=$idvis";
$rs=mysqli_query($link,$sql);
if(mysqli_num_rows($rs)>0)
    {
    $row_xray=mysqli_fetch_array($rs);
    $fimg=$row_xray['img'];
echo $fimg;echo "<br>";


echo "<IMG src='$fimg'>";
echo "<br>";
echo "Description of image: ".$row_xray['description'];
    }

?>
</p>
</section>
</div>
</html>
```