

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО
МОНІТОРИНГУ ПОКАЗНИКІВ ФІЗИЧНОГО СТАНУ ПАЦІЄНТІВ
SOFTWARE FOR REMOTE MONITORING OF PATIENTS'
PHYSICAL CONDITION

Виконав: студент 4 курсу, групи КНТ-128

Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Волощук Д.О.

(прізвище та ініціали)

Керівник Пархоменко А.В.

(прізвище та ініціали)

Рецензент Поляков М.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Волощука Дмитра Олександровича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для віддаленого моніторингу показників фізичного стану пацієнтів. Software for Remote Monitoring of Patients' Physical Condition

керівник проєкту (роботи) Пархоменко Анжеліка Володимирівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 11 травня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Огляд існуючих апаратних та програмних засобів моніторингу показників фізичного стану пацієнтів. 3. Проєктування програмного забезпечення. 4. Конструювання програмного забезпечення 5. Керівництво лікаря. 6. Керівництво пацієнта.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	Пархоменко А.В., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “20” лютого 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка архітектури системи	4 тиждень	Розділ 2
4	Розробка програмного забезпечення	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент

(підпис) Волощук Д.О.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Пархоменко А.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 88 с., 30 рис., 7 табл., 2 дод., 55 джерел.

НОСИМИЙ ПРИСТРІЙ, ПОКАЗНИКИ ФІЗИЧНОГО СТАНУ, СИСТЕМА МОНІТОРИНГУ, ДАТЧИК ПУЛЬСУ, ВЕБЗАСТОСУНОК, ІНТЕРФЕЙС.

Об'єкт дослідження – віддалений моніторинг показників фізичного стану здоров'я пацієнтів.

Предмет дослідження – програмні та апаратні засоби віддаленого моніторингу показників фізичного стану здоров'я.

Мета роботи – розробка вебзастосунку для віддаленого моніторингу показників фізичного стану здоров'я пацієнтів, що надасть інформацію лікарям в реальному часі та дозволить підвищити оперативність надання медичної допомоги.

Матеріали, методи та технічні засоби: мова гіпертекстової розмітки HTML, каскадні таблиці стилів CSS, мова програмування JS, інтегроване середовище розробки JetBrains WebStorm, редактор вихідного коду Microsoft Visual Studio Code, локальний сервер Open Server.

Результати. Створено вебзастосунок, що дозволяє відображати інформацію про пацієнтів та виміряні показники їх фізичного стану в реальному часі.

Висновки. Використання розробленого вебзастосунку дозволить підвищити оперативність надання медичної допомоги та ефективність лікування пацієнтів.

Галузь використання – медичні заклади, в яких потрібно постійно слідкувати за станом здоров'я пацієнтів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 30 figures, 7 tables, 2 appendixes, 55 sources.

WEARING DEVICE, PHYSICAL STATUS INDICATORS, MONITORING SYSTEM, PULSE SENSOR, WEB APPLICATION, INTERFACE.

Object of study is a remote monitoring of physical health indicators of patients.

Subject of study are software and hardware for remote monitoring of patients' physical health indicators.

The goal of the work is to develop a web application for remote monitoring of patients' physical health indicators, which will provide information to doctors in real time and will increase the efficiency of medical care.

Materials, methods, and tools: HTML hypertext markup language, cascading CSS style sheets, JS programming language, JetBrains WebStorm integrated development environment, Microsoft Visual Studio Code source code editor, local server Open Server.

Results. The web application has been created that allows to display information about patients and measured indicators of their physical condition in real time.

Conclusions. The usage of the developed web application will increase the efficiency of medical care and the effectiveness of patient treatment.

The fields of application are medical institutions that need to monitor the indicators of patients' health constantly.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд основних показників здоров'я людини.....	10
1.2 Аналіз сучасних інформаційних технологій в медицині.....	12
1.3 Дослідження існуючих медичних інформаційно-пошукових систем.....	14
1.4 Висновки до розділу 1.....	17
2 Огляд існуючих апаратних та програмних засобів моніторингу показників фізичного стану пацієнтів.....	18
2.1 Сучасні носимі пристрої з функцією вимірювання показників фізичного стану людини.....	18
2.2 Огляд існуючих систем моніторингу.....	22
2.3 Порівняльний аналіз та огляд компонентів носимого пристрою	26
2.4 Висновки до розділу 2.....	30
3 Проєктування програмного забезпечення.....	31
3.1 Аналіз вимог.....	31
3.2 Розробка архітектури системи.....	33
3.3 Вибір інструментарію розробки	35
3.4 Етапи реалізації проєкту.....	43
3.5 Висновки до розділу 3.....	46
4 Конструювання програмного забезпечення.....	47
4.1 Алгоритми функціонування системи.....	47
4.2 Програмна реалізація вебзастосунку.....	50
4.3 Висновки до розділу 4.....	54
5 Керівництво лікаря.....	55

5.1 Призначення й умови застосування програми.....	55
5.2 Характеристики програми.....	55
5.3 Звернення до програми.....	56
5.4 Вхідні та вихідні дані.....	59
5.5 Повідомлення лікарю.....	59
6 Керівництво пацієнта.....	60
6.1 Призначення програми.....	60
6.2 Умови виконання програми.....	60
6.3 Робота з програмою.....	60
Висновки.....	64
Перелік джерел посилання.....	65
Додаток А Текст програми.....	71
Додаток Б Слайди презентації.....	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AJAX	– Asynchronous Javascript and XML;
API	– Application Programming Interface;
HTML	– Hyper Text Markup Language;
ЕКГ	– електрокардіограма;
ІТ	– інформаційні технології;
ОС	– операційна система;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
ТЗ	– технічне завдання;
ЧСС	– частота серцевих скорочень.

ВСТУП

Галузь охорони здоров'я останнім часом зазнала істотних змін завдяки впровадженню сучасних інформаційних технологій. Лікарі активно використовують комп'ютерну техніку та спеціалізоване програмне забезпечення. Відповідні зміни відбуваються на різних етапах надання медичних послуг, а також підготовки медичних фахівців.

Медична інформаційна система Helsi вже активно працює в Україні та дозволяє спростити процес пошуку лікаря, запису на прийом, зберігання важливої інформації щодо обстежень пацієнтів.

Але розвиток телемедицини, тобто надання медичних послуг онлайн, поки потребує активних рішень. Важливою задачею залишається віддалений моніторинг в реальному часі показників фізичного стану пацієнтів (артеріальний тиск, частота серцевих скорочень, насиченість крові киснем, температура, тощо), які можуть знаходитися як в стаціонарі, так і лікуватися вдома. Організація моніторингу цих показників дозволить підвищити ефективність контролю за станом здоров'я пацієнтів та надасть оперативну інформацію лікарям. Для цього потрібно надягнути спеціальний носимий пристрій пацієнту, організувати передачу даних з нього до медичної інформаційної системи, а лікарю авторизуватися в системі для отримання інформації про пацієнта. На жаль, поки таких систем в Україні немає, тому тема роботи є актуальною.

Метою роботи є розробка вебзастосунку для віддаленого моніторингу показників фізичного стану здоров'я пацієнтів, що надасть інформацію лікарям в реальному часі та дозволить підвищити оперативність надання медичної допомоги.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБСЛАСТИ

1.1 Огляд основних показників здоров'я людини

Здоров'я – властивість людини виконувати свої біосоціальні функції в мінливому середовищі, з перевантаженнями і без втрат, за відсутності хвороб і дефектів [1].

У нашому організмі дванадцять систем. Кожна з них – дихальна, травна, ендокринна та інші – мають свій ключовий показник. До основних відносяться: артеріальний тиск, кількість дихальних рухів, частота серцевих скорочень, температура тіла, показники крові та ін. [2].

Функціональний стан організму відбивається в інтенсивності та стійкості роботи органів і систем, що дуже важливо для оцінки здоров'я. Практично будь-який фізіологічний показник може бути використаний для оцінки функціонального стану та здоров'я пацієнта. Однак їх інформативність неоднакова [3].

Показники, що визначають стан здоров'я людини, можливо представити у вигляді декількох груп (рис. 1.1):

- фізіологічні (температура тіла, частота пульсу, артеріальний тиск);
- антропометричні (вага, зріст, об'єм грудної клітки, геометрична форма органів і тканин);
- біохімічні (вміст хімічних елементів в організмі, еритроцитів, лейкоцитів, гормонів та ін.);
- біологічні (відсутність або наявність вірусних та інфекційних хвороб, склад кишкової флори) [4].

Таким чином, це основні показники, які дозволяють людині відчувати себе добре, коли вони в фізіологічній нормі. Тому, їх необхідно контролювати як самостійно, так і у медичних закладах.



Рисунок 1.1 – Класифікація показників фізичного стану людини

Пульс – це вібрація при ударі серця, що призводить до коливань стінок кровоносних судин. Частота серцевих скорочень (ЧСС) – кількість скорочень, що робить серце за хвилину [5]. ЧСС - це оптимальна швидкість просування кисню по крові, разом з кров'ю від легких до органу. Це показник роботи нашої серцево-судинної системи, що відповідає, в тому числі, за регуляцію швидкості води в організмі [2]. Нормальна температура тіла – така при якій вода в нашому організмі знаходиться в стабільному стані і забезпечує всі процеси окислення та відновлення [2]. Артеріальний тиск – це та сила, з якою кров здійснює тиск на стінки артерій [6].

Отже, саме показники, що відносяться до фізіологічних, на сьогоднішній день можуть моніторитися віддалено для пацієнтів, які знаходяться як на стаціонарному, так і на амбулаторному лікуванні.

1.2 Аналіз сучасних інформаційних технологій в медицині

Розвиток сучасної медицини характеризується суттєвими змінами в медичній теорії та практиці, а також підготовці медичних фахівців завдяки впровадженню інформаційних технологій (ІТ) в різноманітні галузі охорони здоров'я людини [7] - [8].

Ці технології спрямовані на допомогу лікарю в діагностиці захворювань, а також збереженні та подальшій обробці отриманої на всіх стадіях лікувального процесу інформації [8].

Згідно з [7], існуючі медичні інформаційні системи можна умовно розділити на такі групи:

- спеціалізовані медичні системи (томографія, рентгенографія, флюорографія, ультразвукове дослідження та ін.);
- діагностичні системи;
- системи електронного документообігу;
- інформаційно-пошукові системи.

Згідно з [8], основними напрямками впровадження ІТ технології в медицині є медична інформаційна система, телемедицина, медична діагностика, експертні системи, медичні апаратно-комп'ютерні системи, робототехніка та ін. (рис. 1.2).



Рисунок 1.2 – Напрямки впровадження ІТ в медицині

Медична інформаційна система представляє собою програмно-апаратний комплекс, що надає можливості збирання, зберігання та подальшої обробки медичної інформації [8].

Телемедицина використовує сучасні телекомунікаційні технології та канали зв'язку для обміну медичною інформацією між фахівцями, а також віддаленого надання медичної допомоги та консультацій в тих випадках, коли відстань і час є критичними факторами [8].

Медична діагностика активно використовує сучасну комп'ютерну техніку при обробці великих об'ємів даних, які отримуються під час досліджень (магніто-резонансна томографія, комп'ютерна томографія, ультразвукографія та ін.) [8].

Експертні системи активно використовуються в сфері медичної діагностики захворювань. Вони дозволяють накопичувати знання фахівців в конкретних предметних галузях і використовувати їх під час моніторингу перебігу захворювань, при прогнозуванні перебігу захворювань й ускладнень, а також плануванні лікувально-діагностичного процесу [8].

Медичні апаратно-комп'ютерні системи дозволяють здійснювати тривалий моніторинг стану хворих на основі довготривалого і безперервного аналізу великого об'єму показників фізичного стану пацієнта (електрокардіограма (ЕКГ), тиск, ЧСС, температура, вміст газів в крові, тощо). До них також відносяться автоматизовані системи інтенсивної терапії, протези, імпланти, штучні органи, що створюються на основі мікропроцесорів та багато інших [8].

Робототехніка – представлена різноманітними роботами (хірурги, манекени, кур'єри, санітари), які виконують певні маніпуляції. Наприклад, медичні роботи ARES, Da Vinci, або медична рукавичка Robotic Glove, яка збирає інформацію з датчиків вібрації, звуку, температуру, має акселерометр і систему оповіщення, що сигналізує лікарю про певні стани та події, що вимагають його уваги [8].

1.3 Дослідження існуючих медичних інформаційно-пошукових систем

Як показали проведені дослідження, на сьогоднішній день частина медичних закладів України мають свої вебсайти, які є більш інформативними, ніж функціональними та містять статичну інформацію.

Прикладом є вебсайт однієї з найпопулярніших та найкращих приватних клінік України, що знаходиться в м. Київ, «БОРИС» (рис. 1.3) [9].

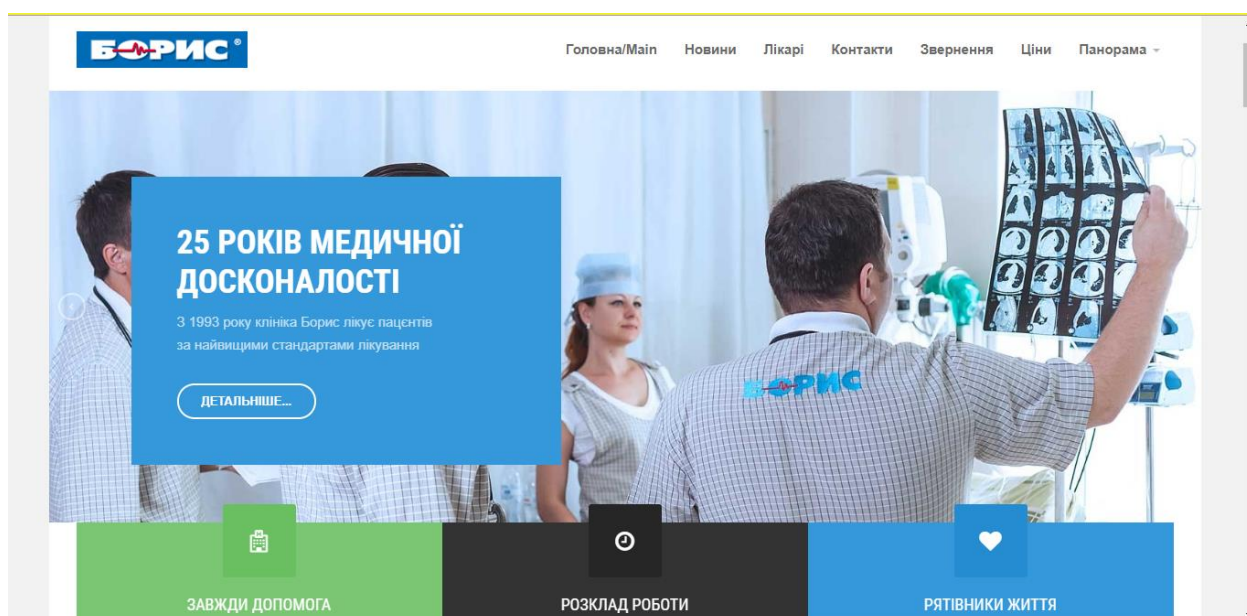


Рисунок 1.3 – Вигляд головної сторінки вебсайту лікарні «БОРИС» [9]

Головну сторінку лікарні «Охмадит» (м. Київ) можливо переглянути на рисунку 1.4 [10].

Головну сторінку лікарні «Святого Миколая», (м Запоріжжя) можливо переглянути на рисунку 1.5 [11].

Аналіз структури та функціональних можливостей вебсайтів показав, що вони відображають інформацію про послуги, ціни, лікарів, а також про місцезнаходженні лікарні. Вебсайт клініки «Святого Миколая» має опцію Особистий кабінет, що дозволяє відстежувати записи пацієнта до лікарів.

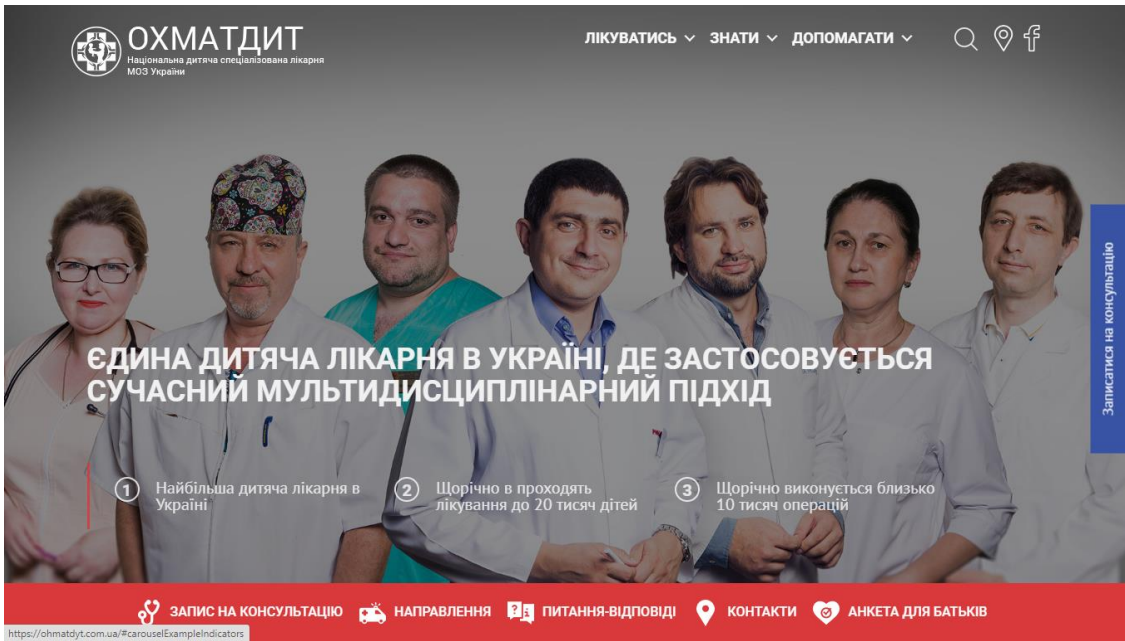


Рисунок 1.4 – Зовнішній вигляд головної сторінки лікарні «Охмадит» [10]

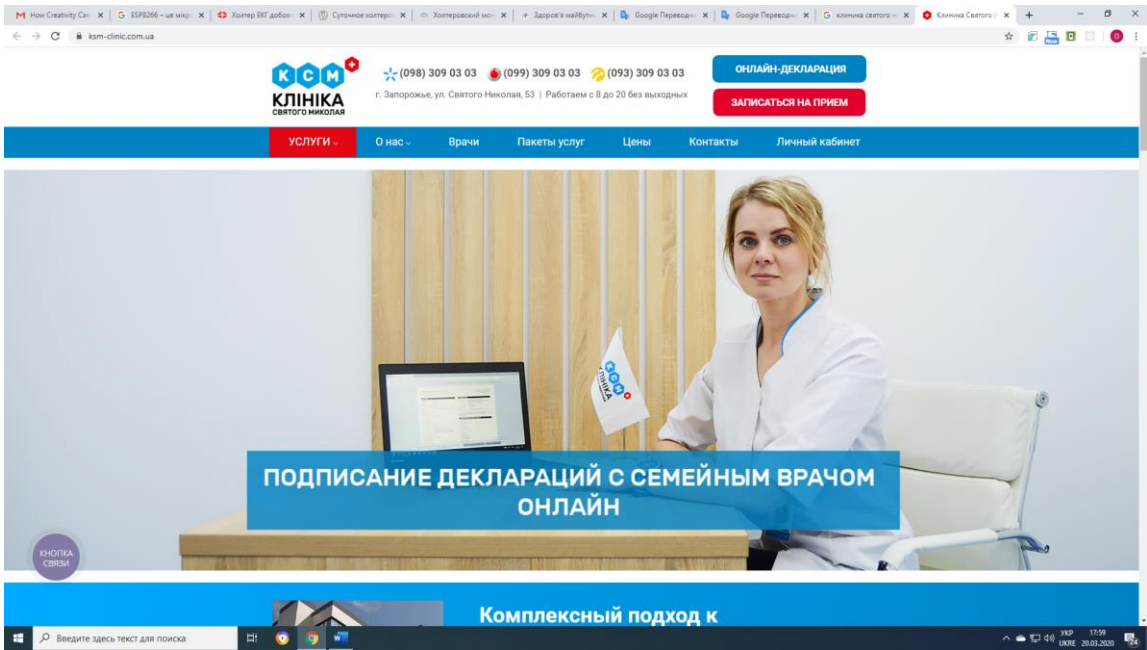


Рисунок 1.5 – Зовнішній вигляд головної сторінки лікарні «Святого Миколая» [11]

Сайти мають гарне оформлення та інформативність, мають яскравий інтерфейс та привабливі ефекти. Вони надають також функцію зворотнього

зв'язку, можливість запису онлайн до лікаря. На видному місці розташовані посилання до соціальних мереж.

Вебсайт клініки «Святого Миколая» має лише логотип українською мовою, але сайт можна використовувати лише на російській мові. Вебсайт лікарні «Охмадит» має інтерфейс українською мовою.

Нещодавно в Україні після медичної реформи було запущено всеукраїнський вебзастосунок HELSI, для запису на прийом до сімейного лікаря, пошуку вузьких фахівців та ін. (рис. 1.6). В даному вебзастосунку можливо знайти свого сімейного лікаря та записатися на прийом на вільний час, що є досить зручним, адже не потрібно більше стояти в черзі. Також, в даному веб застосунку можливо забронювати ліки та забрати своє замовлення в аптеці без черги [12].

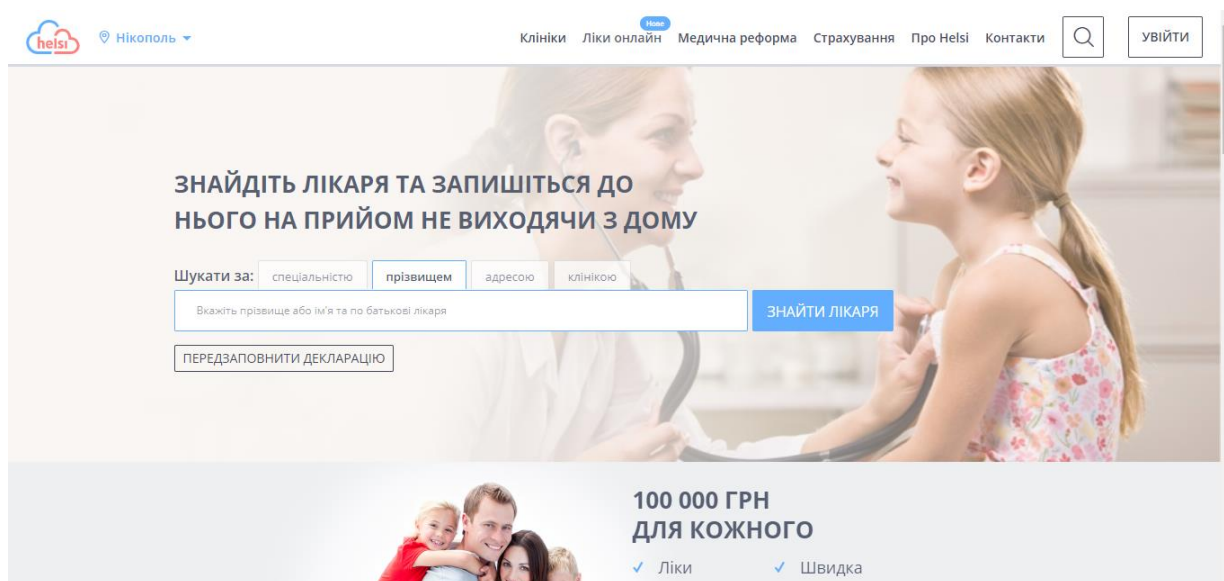


Рисунок 1.6 – Зовнішній вигляд головної сторінки вебзастосунку «HELSI» [12]

1.4 Висновки до розділу 1

В даному розділі було проаналізовано основні фізіологічні показники фізичного стану людини, які можливо контролювати віддалено та забезпечувати лікарів оперативною інформацією про пацієнта в реальному часі.

Виявлено, що використання нових інформаційних технологій у сучасних медичних центрах дозволяє вести облік наданих послуг, зданих аналізів, виписаних рецептів, електронних амбулаторних карт та історій хвороб, а також складати звіти та вести медичну статистику. Серед основних тенденцій, які отримали розвиток останнім часом в розвинених країнах, слід зазначити активне використання можливостей телемедицини та засобів віддаленого моніторингу показників фізичного стану пацієнтів. Але, огляд медичних інформаційних систем в Україні показав, що здебільшого їх функціонал обмежується лише наданням певної інформації та функцій запису до лікаря. Тому, тема роботи є актуальною та пов'язана з подальшим розвитком телемедицини в Україні.

Мета роботи – розробка вебзастосунку для віддаленого моніторингу показників фізичного стану здоров'я пацієнтів, що надасть інформацію лікарям в реальному часі та дозволить підвищити оперативність надання медичної допомоги.

Завдання:

- аналіз та вибір апаратних засобів для вимірювання показників фізичного стану пацієнтів та їх передачі на сервер;
- аналіз вимог до розроблюваного програмного забезпечення та вибір інструментарію розробки;
- розробка вебзастосунку та перевірка його роботи на локальному сервері;
- розробка програмної документації.

2 ОГЛЯД ІСНУЮЧИХ АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ МОНІТОРИНГУ ПОКАЗНИКІВ ФІЗИЧНОГО СТАНУ ПАЦІЄНТІВ

2.1 Сучасні носимі пристрої з функцією вимірювання показників фізичного стану людини

Сьогодні носимі пристрої широко використовуються в медицині, спорті та сфері безпеки. Вони можуть виявляти аномальні та непередбачені ситуації, а також контролювати фізіологічні параметри та симптоми. Ця технологія змінила підхід до охорони здоров'я, оскільки дозволяє здійснювати безперервний моніторинг за пацієнтом без його госпіталізації. Наприклад, через трекер активності медичним працівникам може бути надана інформація про температуру тіла пацієнта, ЧСС, активність мозку, рухи м'язів та інші дані. [13].

На сьогоднішній день існує велика кількість приладів, які дозволяють вимірювати ЧСС. Найпопулярнішими є розумні годинники, фітнес трекери та пульсометри.

Фітнес-трекер (фітнес-браслет, трекер активності) – пристрій призначений для моніторингу таких показників як: пройдена відстань, споживання калорій, показники серцевого ритму та якості сну. Фітнес трекер замінює: секундомір, крокомір, годинник, будильник, пульсометр, спідометр, висотомір. В даний час цей термін найчастіше використовується для розумного годинника, який синхронізується зі смартфоном або комп'ютером для відстеження даних, і виконує роль трекера активності. Також існують мобільні програми для відстеження фітнес активності [13].

Одним з найпопулярніших фітнес-трекерів є Mi Band, а його зовнішній вигляд можливо переглянути на рисунку 2.1.



Рисунок 2.1 – Зовнішній вигляд фітнес-трекеру «Mi Band» [13]

Пульсометри – пристрій персонального моніторингу ЧСС в реальному часі або його запису для подальшого дослідження. Широко використовується у тренуваннях та змаганнях любителями та спортсменами циклічних видів спорту, таких як бігові види легкої атлетики (особливо біг по шосе), лижні гонки, велоспорт, плавання [14]. Зовнішній вигляд пульсометра Sigma Sport PC 15.11 зображено на рисунку 2.2.



Рисунок 2.2 – Зовнішній вигляд пульсометра Sigma Sport [15]

В Україні часто сімейні лікарі використовують на своїх прийомах пульсоксиметри. На рисунку 2.3 зображено приклад такого пульсоксиметра [16].

Пульсоксиметр – це невеликий пристрій, який безболісно та швидко вимірює рівень насичення киснем, що міститься в еритроцитах, або кількість

кисню в крові. Ці пристрої мають зонд або датчик, який виглядає як затискач для пластикового пакета і кріпиться на кінчик пальця або мочку вуха [17].

На маленькому екрані на зонді відображається відсоток, який визначає, скільки кисню в крові знаходиться у кінцівках або інших частинах тіла пацієнта [17].

Пандемія COVID-19 призвела до поновлення інтересу до пульсоксиметрів для домашнього використання. Перевірка того, чи рівень насичення киснем перевищує рекомендований рівень, може допомогти визначити, чи потрібна невідкладна медична допомога. Такі випадки можуть бути викликані гострими захворюваннями, такими як пневмонія, або спалахами хронічних захворювань, таких як астма. Оксиметри також можуть бути корисними для екстремальних видів спорту [17].



Рисунок 2.3 – Зовнішній вигляд пульсоксиметру [16]

Багато розумних годинників можуть виконувати функції фітнес-трекера, як мінімум порахувати пройдені кроки, відстань і кількість витрачених під час тренування калорій. Багато пристроїв обладнані також датчиком ЧСС, можуть автоматично визначати початок і кінець пробіжки, або тренування, і навіть підрахувати кількість підходів для кожної вправи [18].

Найпопулярніший в світі розумний годинник - це Apple Watch. Остання версія цього годинника може зробити ЕКГ. Зовнішній вигляд Apple Watch зображено на рисунку 2.4 [19].



Рисунок 2.4 – Зовнішній вигляд Apple Watch [19]

Виявлені переваги та недоліки розумних годинників та фітнес-трекерів наведено в таблиці 2.1.

Таблиця 2.1 – Переваги та недоліки фітнес-трекерів та розумних годинників

	Фітнес-трекер	Розумний годинник
Переваги	Вартість нижче, ніж у годинника; розроблені для любителів спорту та фітнесу; зручні та легкі; широкий вибір колірної гами та розмірів; безліч функцій для відстеження активної діяльності; можуть виглядати як годинник.	Величезна кількість функцій; витончений та стильний дизайн; з'єднуються зі смартфоном
Недоліки	Обмежений функціонал; не є заміною медичного приладу для вимірювання життєво важливих показників	Безліч функцій може спантеличити користувача; вага та габарити більше, ніж у фітнес-трекера; невеликий термін життя батареї; висока вартість.

2.2 Огляд існуючих систем моніторингу

Дослідження показали, що в роботах [20] - [27] представлені результати проєктування різних систем моніторингу для сфери охорони здоров'я. Усі вони мають специфічні особливості реалізації, переваги, недоліки та обмеження.

Автори роботи [20] пропонують рішення для моніторингу стану вагітних у стаціонарі. Система заснована на використанні семи ретрансляційних вузлів для охоплення внутрішнього середовища, енергоефективного протоколу множинного доступу з тимчасовим розділенням (Time Division Multiple Access, TDMA) та механізму аварійного порятунку за допомогою SMS/e-mail. Але площа охоплення системи складає всього 50x20 метрів, що недостатньо для покриття території лікарні.

Представлена в [21] система заснована на використанні смартфона, який може розрядитися або який пацієнт може забути в лікарняній палаті, коли вийде на прогулянку (рис. 2.5).



Рисунок 2.5 – Архітектура системи моніторингу [21]

Запропонована в [22] бездротова амбулаторна система моніторингу плода збирає сигнали ЕКГ та електроміограми з поверхні шкіри пацієнта, а потім перетворює дані Bluetooth у сигнали правильного формату для введення на фетальний монітор. Ця система має обмежений діапазон через використання технології Bluetooth.

У [23] запропонована архітектура системи дистанційного моніторингу здоров'я, яка об'єднує пристрої домашньої автоматизації, а також пристрої моніторингу показників стану мешканців будинку (пульс, температура тіла, артеріальний тиск, рівень глюкози в крові, ЕКГ). Автори зосереджені на аналізі та виборі протоколу CoAP, що дозволяє зменшити вимоги до пропускну здатності та обсягу генерованих даних. Проте організація вимірювань життєвих показників мешканців будинку не описана.

Бездротова система моніторингу серцевого ритму та температури на основі використання плати Arduino та радіомодуля RF24L01 запропонована в [24]. За допомогою цього модуля можна підключити максимум сім пристроїв, що часто є недостатнім. Крім того, більшість виробників модулів nRF24L01 з невеликою потужністю передавача вказують відстань його дії всього 100 метрів. Також можуть виникнути проблеми з живленням модуля, оскільки в модулі живлення 3,3 В не передбачена необхідна сила струму. Більшість проблем з передачею виникає саме через шум від джерела живлення 3,3 В. Дані з датчиків відображаються на локальному РК-дисплеї, підключеному до Arduino.

Автори [25] пропонують варіант дешевої системи моніторингу серцебиття та температури для віддалених пацієнтів. Система заснована на використанні двох мікроконтролерів PIC16F72, а також радіочастотного передавача і приймача для вимірювання, передачі та прийому даних на частоті 433 МГц. Результати відображаються на РК-дисплеї. Радіус дії 15 метрів.

Автори роботи [26] пропонують використовувати стаціонарний пристрій на базі мікроконтролера PIC16F877A для збору даних з датчика температури та пульсу, а також відображення їх на РК-дисплеї для пацієнта. Дані передаються через Інтернет на персональну сторінку користувача. Лікар може переглядати ці дані і в екстрених випадках повідомлення надсилається

через підключений GSM-модем на його мобільний телефон. Медсестра також оповіщується за допомогою звукового сигналу.

Результати розробки монітора серцебиття та температури тіла на основі мікроконтролера PIC16F73 представлені в [27]. Автори проаналізували та рекомендували два варіанти модулів (RF або GSM) для передачі даних, які відображаються на мобільному пристрої у вигляді SMS, що є недостатньо інформативним.

В роботі [28] запропоновано підхід до дистанційного моніторингу показників серцевого ритму пацієнтів стаціонару. У проєкті використовувався метод підключення до мережі на основі кількох точок доступу та мережі Mesh (рис. 2.6), а також сервіс Freeboard.io IoT, що дозволяє зберігати та візуалізувати інформацію про стан пацієнтів у лікарні.

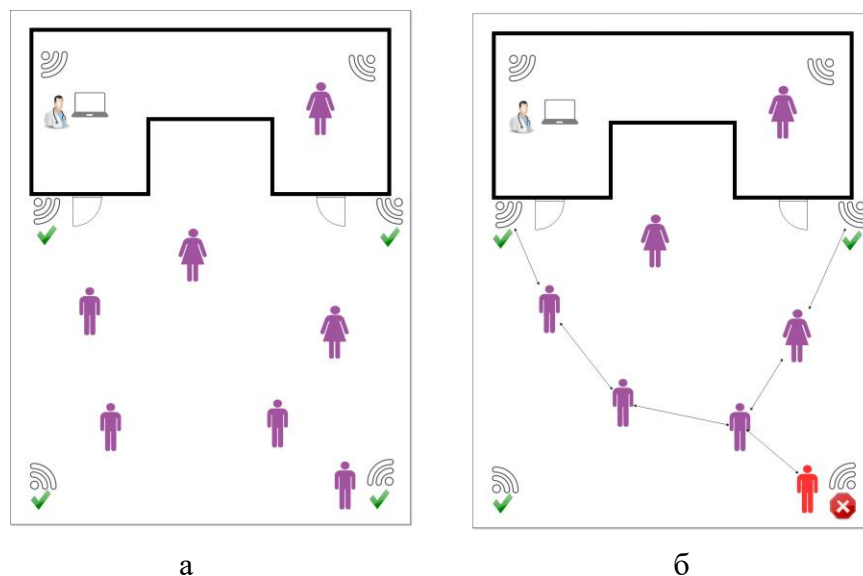


Рисунок 2.6 – Варіанти топології мережі на основі кількох маршрутизаторів (а) та мережі типу Mesh (б) [28]

Розроблений в [28] прототип програмно-апаратного комплексу для моніторингу ЧСС (рис. 2.7) заснований на використанні Wi-Fi модуля WeMOS D1 Mini ESP8266, який передає дані на сервер, а п'єзоелектричний датчик сповіщає пацієнта про те, що пульс перевищує норму. Структура прототипу представлена на рисунку 2.8.

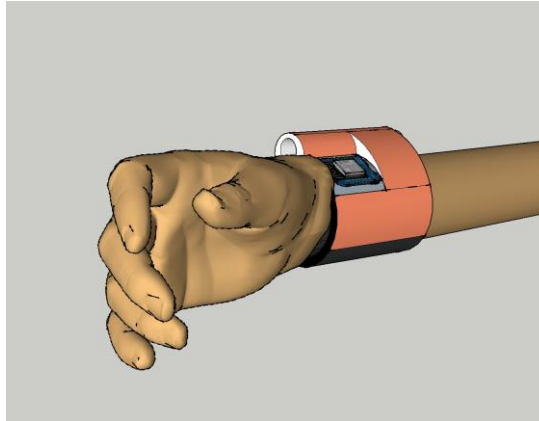


Рисунок 2.7 – Прототип носимого пристрою [28]

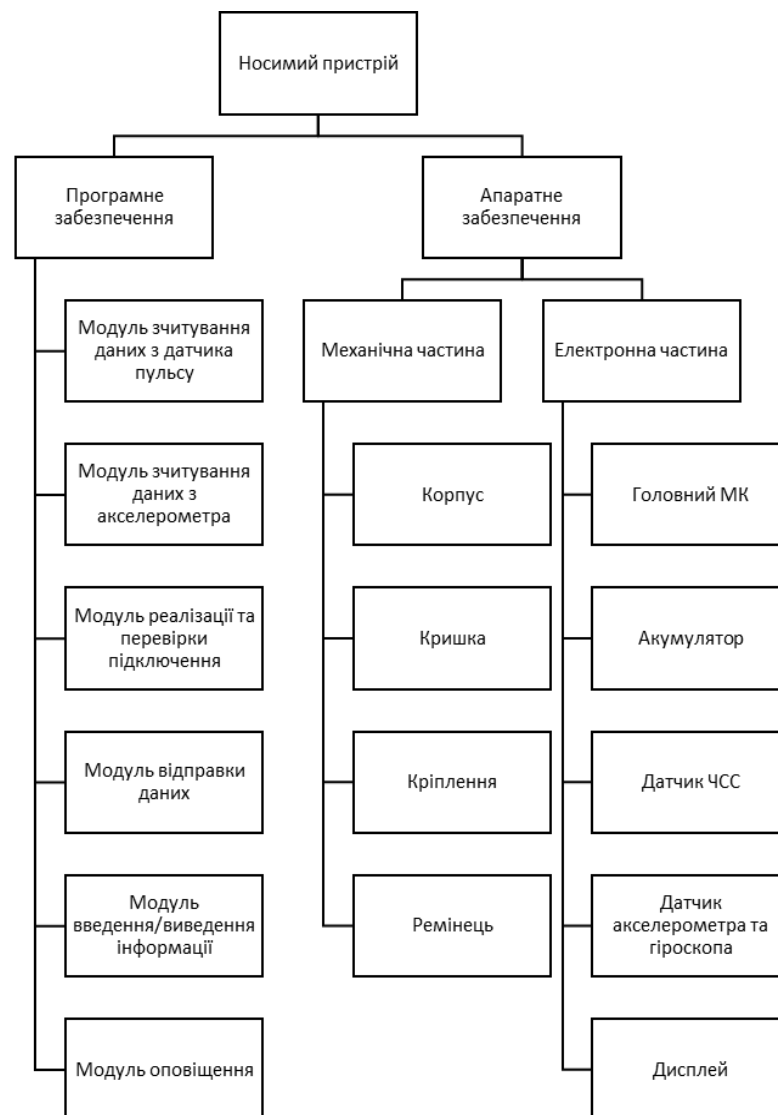


Рисунок 2.8 – Структура прототипу носимого пристрою [28]

Таким чином, у представлених роботах для спостереження за станом пацієнта здебільшого використовуються стаціонарні пристрої. Лише в роботах [21] та [28] основу моніторингової системи складають носимі пристрої.

У більшості випадків покриття, що забезпечує система не відповідає завданням навігації для пацієнтів, які можуть переміщуватися по території лікарні. У деяких випадках поточні дані відображаються тільки на локальному дисплеї, без підключення до персонального комп'ютера (ПК). Використання смартфонів також не є оптимальним, оскільки пацієнт може забути його в палаті, загубити або вчасно не зарядити. У представлених роботах (за винятком роботи [28]) не використовуються можливості сучасних хмарних технологій для зберігання та візуалізації великої кількості даних.

2.3 Порівняльний аналіз та огляд компонентів носимого пристрою

Основними складовими апаратної частини носимого пристрою [28] є Wi-Fi модуль та датчик ЧСС.

Вибір Wi-Fi модуля базується на аналізі популярних модифікацій Wi-Fi модуля ESP8266. Порівняльний аналіз кількох найпоширеніших версій представлено в таблиці 2.2. На основі цих модулів створено багато плат для розробників: NodeMCU, Wemos, LoLin V3, Witty Cloud, LuaNode32, HUZZAH ESP.

Плата Wemos є аналогом плати Witty Cloud і NodeMCU v3, яка має найменші розміри і використовує версію ESP-12F модуля ESP-8266 (рис. 2.9) [29]. WeMos D1 mini не потребує зовнішнього мікроконтролера або іншого пристрою керування, оскільки, крім модуля Wi-Fi, ESP-12F вже має 32-розрядний мікроконтролер з тактовою частотою 80 МГц, а також чіп флеш-пам'яті на 4 Мб [28].



Рисунок 2.9 – Зовнішній вигляд плати WeMos D1 [29]

Таблиця 2.2 – Порівняння модифікацій модуля ESP8266 [28]

	Кількість GPIO виводів	Антенa	АЦП	Bluetooth	Вартість, \$
ESP-01	3	-	-	-	3
ESP-07	9	+	-	-	2.97
ESP-12F	12	+	1	-	3.04
ESP32	32	+	14	+	6.72

Міні-модуль WeMos D1 є гарним рішенням для застосунків IoT, систем віддаленого моніторингу або керування, автономних датчиків тощо. Плата пропонує кілька варіантів роботи з мережами Wi-Fi, вона може виступати в ролі клієнта Wi-Fi мережі або створити точку доступу. Використання WeMos D1 mini замість модуля ESP8266 значно спрощує конструкцію, оскільки в модуль вже вбудований конвертер CH340 USB-UART і регулятор напруги, контакти розведені до роз'єму зі стандартним кроком 2,54 мм. На цій платі також передбачено розширення функціональності за допомогою допоміжних модулів, що дозволяє додати кнопку тривоги, дисплей тощо [28].

Основними показниками продуктивності розробленої системи дистанційного моніторингу є охоплення, споживання енергії та час затримки. Експериментально, діапазон взаємодії між модулями WeMos D1 mini був встановлений для підключення «точка-точка». Відстань становила 40 метрів на відкритому просторі з перешкодами у вигляді дерев за допомогою антен РСВ. Враховуючи зону покриття точки доступу Wi-Fi Ubiquiti UniFi AP (150 метрів без підключення зовнішньої антени) і використану топологію, в результаті можна охопити територію лікарні близько 600х600 метрів [28].

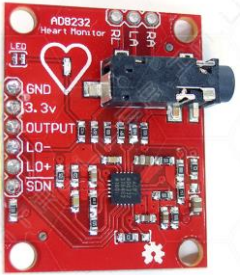
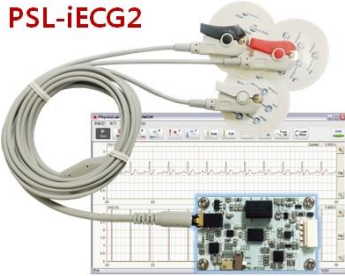
Споживання енергії міні-модулем WeMos D1 коливається від 20 мкА до 250 мА. Споживання датчика імпульсів становить 4 мА, а для випробувань використовувалися акумулятори типу 18650 ємністю 2400 мА. Враховуючи безперервний моніторинг імпульсної та непотокової (пакетної) тимчасової передачі даних, розрахунковий час роботи системи становить 15 годин. Вартість носимого пристрою становить приблизно 50 доларів [28].

В якості датчика імпульсу в конструкції прототипу був обраний недорогий Pulse Sensor для Arduino (рис. 2.10) [30] - [31]. Для підвищення точності вимірювань можна рекомендувати інші датчики, що представлені в таблиці 2.3.



Рисунок 2.10 - Pulse Sensor для Arduino [30]

Таблиця 2.3 – Порівняння популярних датчиків серцевого ритму та пульсоксиметрії [28]

Назва датчику пульсу	Зовнішній вигляд	Тип вимірювання	Розміри, мм
KY-039 [32]		Оптичний	26x24x10
CJMCU [33]			15,875x3,175 - діаметр/товщина
MAX 30100 [34]			5,6x2,8x1,2
AD8232 [35]		Контактний	35x28 + 3 контакти зі з'ємними електродами
PSL-iECG2 [36]			56x35 + 3 контакти зі з'ємними електродами

2.4 Висновки до розділу 2

В даному розділі були оглянуті основні можливості існуючих носимих пристроїв з функціями контролю показників фізичного стану пацієнтів – розумних годинників, фітнес-трекуерів та пульсоксиметрів. Показано, що ці пристрої складно інтегрувати в систему моніторингу лікарні, через функціональні особливості та високу вартість (розумні годинники).

Існуючі системи моніторингу здебільшого будуються на стаціонарних, а не на носимих пристроях, які дають цілий ряд переваг.

За основу вирішено взяти прототип носимого пристрою з роботи [28], що побудований на основі Wi-Fi модуля WeMos D1 та датчика Pulse Sensor для Arduino. Актуальною є задача розробки вебзастосунку, в який будуть передаватися дані з цього носимого пристрою для їх збереження та візуалізації.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз вимог

Вебзастосунок призначений для моніторингу лікарями показників фізичного стану пацієнтів, що використовують носимі пристрої на основі Wi-Fi модуля WeMos D1 та датчика Pulse Sensor.

До основних вимог до вебзастосунку відносяться наступні:

- наявність функцій додавання, збереження та видалення пацієнтів;
- відображення показників фізичного стану кожного пацієнта, що отримуються за допомогою носимого пристрою;
- здійснення входу до адміністративної частини лікарям лише за логіном та паролем;
- швидкий та зручний пошук інформації.

Вимоги до додавання пацієнтів:

- наявність полів для введення імені та прізвища;
- наявність поля ідентифікатора носимого пристрою;
- наявність поля номера палати пацієнта.
- наявність перевірки полів на заповнення.

Вимоги до відображення всіх пацієнтів:

- відображення всіх доданих пацієнтів;
- відображення імені, прізвища пацієнта;
- відображення номера носимого пристрою та номера палати пацієнта;
- відображення кнопки видалення пацієнта.

Вимоги до авторизації:

- наявність полів для введення логіну та паролю;
- наявність кнопки «Login»;
- перевірка правильності введених даних;

- авторизація та надання доступу до адміністративної панелі лише лікарям.

Даний програмний продукт має функціонувати в операційній системі (ОС) Windows, при наявності вебпереглядача. Перевагу було надано ОС Windows 10. Windows 10 – це ОС для різних апаратних платформ, розроблена компанією Microsoft, яка буде підтримувати її до 2025 року.

Для перегляду вебзастосунку можуть використовуватися будь-які вебпереглядачі: Google Chrome, Opera, Microsoft Edge, Mozilla Firefox, Internet Explorer, Safari та інші.

Вимоги до програмних засобів:

- наявність ОС Windows;
- наявність будь-якого вебпереглядача (Google Chrome, Opera, Microsoft Edge, Safari, Mozilla Firefox, Internet Explorer та інші);
- для доповнення, або зміни певної частини вебзастосунку необхідно встановлення будь-якого локального серверу (OpenServer, Denwer) та редактора коду (Visual Studio Code, Sublime Text, Notepad++);
- доступ до мережі інтернет.

Для нормальної роботи вебзастосунку необхідно мати ПК з такими характеристиками:

- процесор з тактовою частотою не менш 1600 МГц;
- оперативна пам'ять не менш 8 Гб;
- відеокарта NVIDIA GeForce 256Mb, або інша з подібними характеристиками;
- наявність вебпереглядача та доступ до Інтернету;
- монітор, клавіатура та маніпулятор «миша».

3.2 Розробка архітектури системи

UML-діаграма процесу віддаленого моніторингу ЧСС у пацієнтів наведена на рисунку 3.1. Усі пацієнти оснащені носимими пристроями (спеціальними браслетами) з датчиками серцебиття та зареєстровані в системі дистанційного моніторингу для контролю показників фізичного стану.

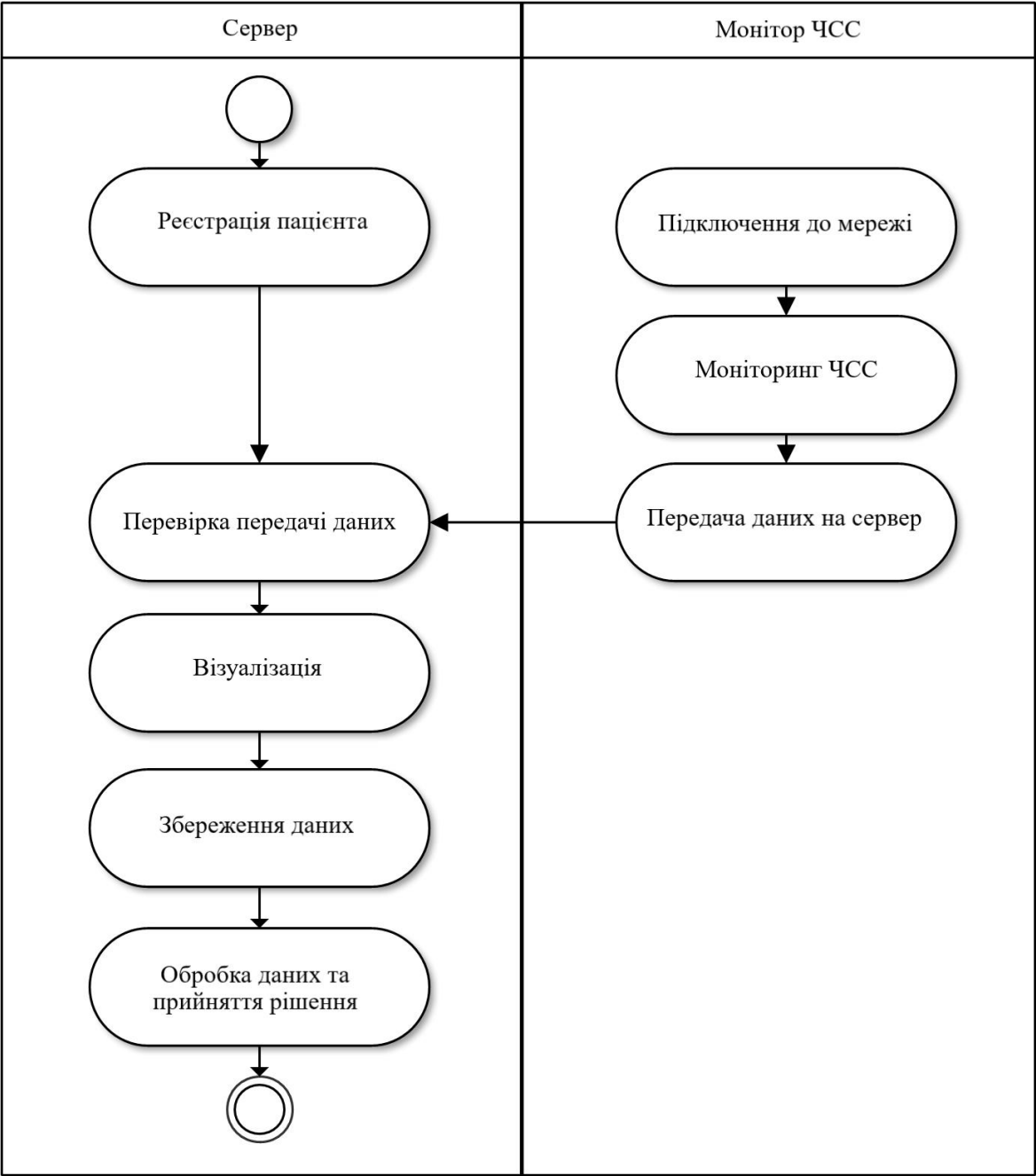


Рисунок 3.1 - UML-діаграма процесу віддаленого моніторингу

Під час моніторингу перевіряється відповідність ЧСС встановленим нормальним значенням. У разі, якщо значення не відповідає нормі, пацієнту та медперсоналу, що знаходяться поруч, подається звуковий сигнал, а на моніторі чергового лікаря з'являється відповідне сповіщення. Всі дані моніторингу серцевого ритму передаються і зберігаються на сервері для подальшої обробки та формування рекомендацій пацієнту. Для цього можливо також застосовувати сучасні IoT сервіси, які гарантують безпечне зберігання та візуалізацію даних.

Узагальнену структурно-функціональну схему вебзастосунку наведено на рисунку 3.2.



Рисунок 3.2 – Структурно-функціональна схема вебзастосунку

Вебзастосунок складатиметься з двох частин – клієнтської (для пацієнтів) та адміністративної (для лікарів). Лікар (адміністратор системи) має можливість додавати, редагувати та видаляти відомості про пацієнтів (ім'я, прізвище, ідентифікатор носимого пристрою, номер палати). Основною

задачею лікаря є перегляд показників фізичного стану пацієнтів в реальному часі (ЧСС, температура, тощо).

Пацієнти мають можливість переглянути інформацію про лікарів та статистику щодо показників свого здоров'я за певний проміжок часу.

3.3 Вибір інструментарію розробки

Як показали проведені дослідження, на сьогоднішній день для розробки вебзастосунків зазвичай використовують:

- мову гіпертекстової розмітки (HyperText Markup Language, HTML);
- каскадні таблиці стилів (Cascading Style Sheets, CSS);
- мультипарадигмальну мову програмування JavaScript;
- технологію AJAX.

Більшість вебсайтів створюється за допомогою мови HTML, яка може бути оброблена браузером та відображена у звичному для користувачів вигляді на екрані комп'ютера [37] - [38].

HTML – це стандартна мова розмітки гіпертексту у Всесвітній павутині, що надає засоби для:

- розробки документу певної структури тексту з використанням заголовків, абзаців, списків, таблиць, тощо;
- використання зображень, аудіо, відео та інших об'єктів додатково до основного тексту;
- отримання даних із мережі інтернет на основі гіперпосилань;
- розробки форм для інтерактивної взаємодії [37].

Для створення стилів сторінок широко використовують CSS, що оброблюють зовнішній вигляд вебсторінки, дозволяють контролювати колір тексту, стиль шрифтів, відстань між параграфами, розміри та розташування колонок, використовувані фонові зображення та кольори, макети дизайну, варіанти відображення на різних пристроях та розмірах екрану та ін. [39].

До основних переваг CSS можна віднести наступні:

- простота обслуговування, що дозволяє змінити стиль, щоб всі елементи на вебсторінках оновилися автоматично;
- покращені стилі для HTML, оскільки CSS має набагато ширший набір атрибутів, ніж HTML, що дає кращий вид HTML-сторінки;
- CSS забезпечують сумісність кількох пристроїв, оскільки дозволяють оптимізувати контент для більш ніж одного типу пристроїв; тобто, використовуючи один і той же HTML-документ, можна забезпечити різні версії вебсайту для різних пристроїв;
- використання CSS у HTML-сторінках дозволяє зробити їх сумісними з сучасними браузерами, оскільки атрибути HTML застаріли, а CSS наближені до глобальних вебстандартів;
- вебзастосунки можуть зберігати CSS локально за допомогою оффлайн-кешу, що надає можливість офлайн перегляду; крім того, кеш забезпечує швидке завантаження та кращу загальну продуктивність вебсайту;
- скрипт забезпечує незалежність від платформи та підтримує всі браузери [39].

Таким чином, HTML та CSS – це дві найбільш фундаментальні технології, які використовуються для створення вебсторінок, підходять для написання сценаріїв та додавання графіки, а також можуть бути використані для створення вебзастосунків. HTML застосовується для розробки структури сторінки, а CSS - для створення макета сторінки на багатьох пристроях та платформах [40].

Для того, щоб зробити вебсторінку інтерактивною, тобто такою що реагує на дії користувача може бути використано мову програмування JavaScript, яка була розроблена на основі мови програмування Java, але JavaScript є зрозумілішою та легшою у використанні ніж Java. JavaScript істотно розширює можливості при розробці вебсторінок та робить їх більш живими та динамічними [41].

Згідно з дослідженнями [40] JavaScript лідирує у рейтингу популярних мов програмування, маючи понад 60% розробників, які її використовують (рис. 3.3).

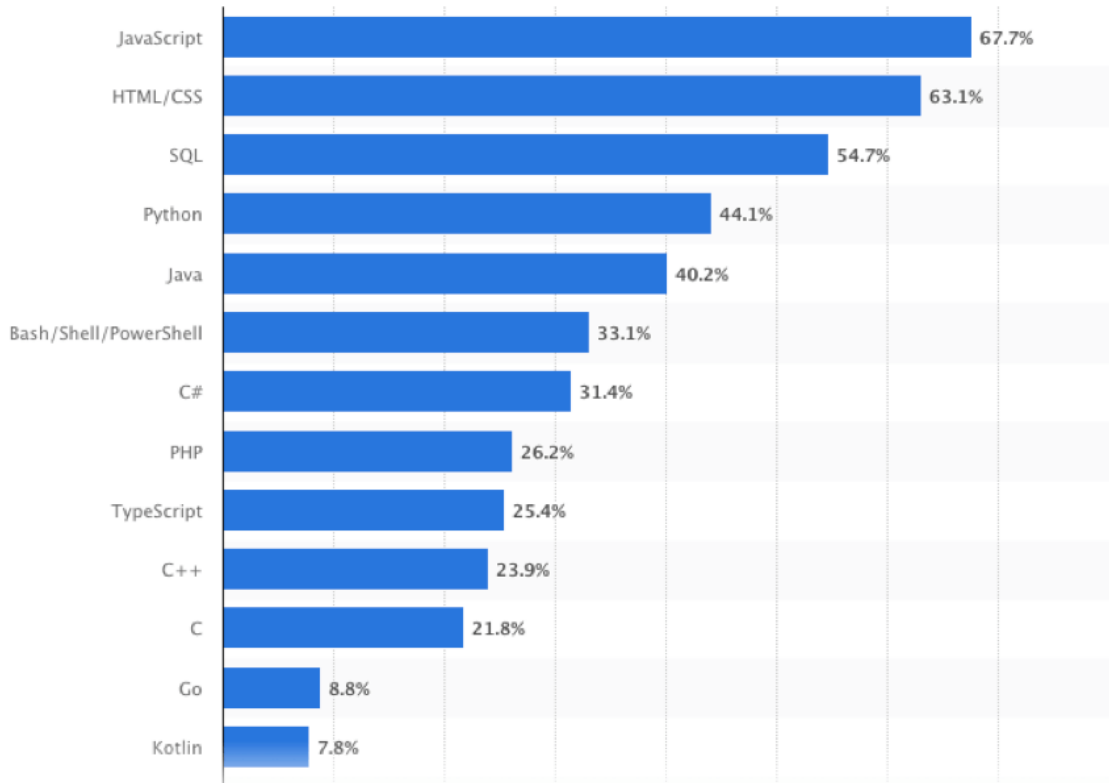


Рисунок 3.3 – Використання популярних мов програмування [40]

В таблиці 3.1 наведено короткий опис деяких популярних мов програмування, що були представлені на рисунку 3.3.

Для кращої роботи JavaScript з HTML може бути використано бібліотеку jQuery.

jQuery – це бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML, яка значно спрощує і прискорює написання JavaScript коду. Вона дозволяє створювати анімацію, обробники подій, що значно полегшує вибір елементів в DOM і створення запитів Asynchronous Javascript and XML (AJAX). Дана бібліотека працює з усіма браузерами. Це означає, що не потрібно турбуватися про кросбраузерність сумісності JavaScript коду [42].

Таблиця 3.1 – Опис популярних мов програмування [40]

Мова	Опис
Javascript	JavaScript – це мова структурованого програмування, що інтерпретується, високорівнева скриптова зі слабкою динамічною типізацією і поліпарадигмальна
HTML/CSS	HTML та CSS – це дві основні технології для створення вебсторінок
Java	Створена компанією Sun Microsystems, Java є об'єктно-орієнтованою та кросплатформною мовою програмування
Python	Python є мовою програмування, яка дозволяє швидко розробляти і більш продуктивно інтегрувати системи
PHP	PHP, він же препроцесор гіпертексту, є скриптовою мовою з відкритим кодом для веброзробки.
C++	C++ – це об'єктно-орієнтована мова програмування з відкритим вихідним кодом
C#	Розроблена MS, ця мова програмування є універсальною.

Для jQuery написана величезна кількість плагінів, які дозволяють розширити її можливості ще більше [43].

jQuery має наступні можливості:

- звертатися до будь-якого елемента об'єктної моделі документа і не тільки звертатися, але і маніпулювати ними;
- працювати з подіями;
- може легко здійснювати різні візуальні ефекти;
- працювати з AJAX, що дозволяє спілкуватися з сервером без перезавантаження сторінки;
- має величезну кількість JavaScript плагінів, призначених для створення елементів призначених для користувача інтерфейсів.

Бібліотека jQuery надає зручний Application Programming Interface (API) для роботи з технологією AJAX для звернення до сервера без перезавантаження сторінки.

За рахунок AJAX зменшується час відгуку і вебсайт по інтерактивності більше нагадує десктоп. Під AJAX мається на увазі будь-яке спілкування з сервером без перезавантаження сторінки, організоване за допомогою JavaScript [44].

AJAX є одним з підходів до розробки користувацьких вебінтерфейсів, який передбачає «фоновий» обмін даними між браузером і серверними ресурсами. Асинхронний спосіб оновлення даних вебсторінок дозволяє не перезавантажувати їх вміст повністю. Завдяки цьому вебресурс працює більш плавно і швидко. Також це дозволяє економити трафік, що особливо актуально в умовах, якщо немає безлімітного інтернету [45].

Дуже поширеним завданням в сучасних вебсайтах і застосунках є отримання окремих елементів даних з сервера для оновлення розділів вебсторінки без необхідності завантаження всієї нової сторінки. Ця особливість суттєво впливає на продуктивність і поведінку сайтів [46].

Модель отримання даних з серверу зображено на рисунку 3.4 [46]. Після відправки запиту на сервер вебсайту, якщо все працює, вся необхідна інформація про вебсторінку завантажувалася та відображалася на комп'ютері.

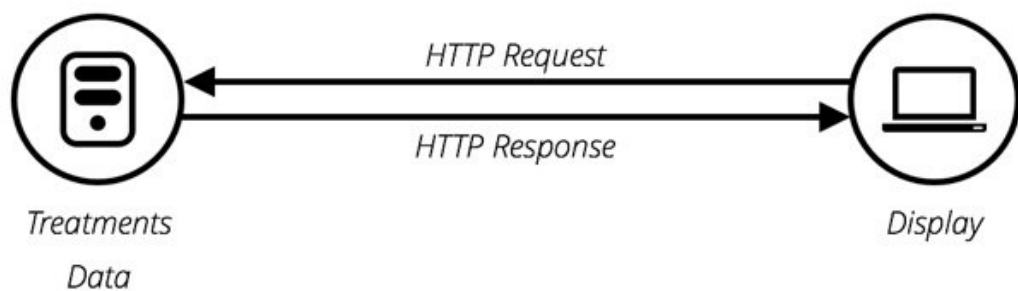


Рисунок 3.4 – Традиційна модель отримання даних з серверу [46]

Але, якщо потрібно оновити будь-яку частину сторінки, доведеться знову завантажити всю сторінку. Це дуже проблематично для складних сторінок та призвело до створення технологій, що дозволяють вебсторінкам

запитувати невеликі фрагменти даних (наприклад, HTML, XML, JSON або звичайний текст) і відображати їх тільки при необхідності, допомагаючи вирішувати проблему [46].

Це досягається за допомогою таких API, як XMLHttpRequest або Fetch API. Ці технології дозволяють вебсторінкам безпосередньо обробляти запити HTTP для певних ресурсів, доступних на сервері, і форматовувати результуючі дані по мірі необхідності перед їх відображенням [46].

Модель AJAX передбачає використання веб-API як проксі для розумнішого запиту даних, а не просто для того, щоб браузер перезавантажував всю сторінку (рис. 3.5) [46].

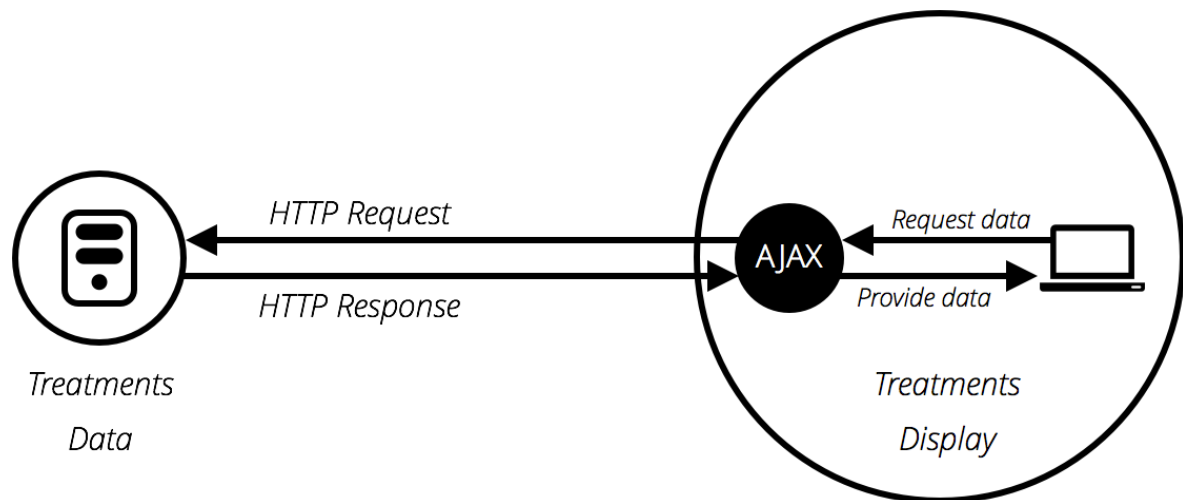


Рисунок 3.5 – Модель отримання даних з серверу на основі AJAX [46]

Переваги застосування технології AJAX:

- оновлення сторінки здійснюється швидше, і не потребує багато часу на її перезавантаження сторінки, відповідно, сайт працює швидше і краще сприймається [46];
- менше даних завантажується при кожному оновленні, що означає менше споживання пропускної спроможності, що особливо важливе для мобільних пристроїв та країн, які не мають повсюдного швидкого інтернет-сервісу [46].

Для розробки вебзастосунку можуть використовуватися різноманітні інтегровані середовища розробки.

Гарне середовище розробки забезпечує:

- компіляцію коду;
- автозакриття тегу;
- перевірку синтаксичних помилок;
- автозавершення;
- налаштування (зміна тем, додавання розширень або плагінів);
- функцію Знайдіть і замініть (як у Microsoft Word);
- ієрархію файлів;
- підключення до репозиторію Git;
- виконання команди терміналу [47].

Результати порівняльного аналізу середовищ веброзробки (PhpStorm, Atom, NetBeans, Sublime Text, Visual Studio Code) наведено в таблиці 3.2 [47].

Таблиця 3.2 – Порівняння середовищ розробки [47]

	PhpStorm	Visual Studio Code	NetBeans	Sublime Text	Atom
Перевірка орфографії	+	+	+	+	+
Підсвічування синтаксису	++	+*	+	+	+
Автозаповнення	++	+	-	+	+
Підтримка Git	+	+	+	+	+
Кросплатформенність	+	+	+	+	+
Оновлюваність	+	+	+	-	+

* – потребує встановлення та розширення

На основі проведеного порівняння, для розробки вебзастосунку було обрано середовище розробки PhpStorm.

PhpStorm – інтегроване середовище розробки для PHP, HTML і JavaScript від компанії JetBrains, розроблене на основі платформи IntelliJ IDEA.

PhpStorm – це інтелектуальний редактор з можливостями аналізу коду на льоту, запобігання помилок у коді і автоматизованими засобами рефакторинга для PHP і JavaScript [48].

Користувальницький інтерфейс цього середовища дуже зручний для користувачів, надає контроль версій, підтримує плагіни та теми [47].

Вебзастосунок буде розроблено за допомогою редактору вихідного коду Visual Studio Code, що розроблений компанією Microsoft для різних ОС [49].

Може використовуватися для кросплатформенної розробки веб і хмарних застосунків. Забезпечує відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense і засоби для рефакторинга. Надає можливості для кастомізації: призначені для користувача теми, поєднання клавіш і файли конфігурації. Розповсюджується безкоштовно, розробляється як ПЗ з відкритим вихідним кодом, але готові збірки розповсюджуються під проприетарною ліцензією [49].

Для обміну даними в системі буде використано JSON (JavaScript Object Notation) – це загальний формат для представлення значень і об'єктів. Спочатку він був створений для JavaScript, але багато інших мов також мають бібліотеки, які можуть працювати з ним. Таким чином, JSON легко використовувати для обміну даними, коли клієнт використовує JavaScript, а сервер написаний на Ruby / PHP / Java або будь-якій іншій мові. JavaScript надає методи: `JSON.stringify` для перетворення об'єктів в JSON та `JSON.parse` для перетворення JSON назад в об'єкт [50].

3.4 Етапи реалізації проєкту

Основні етапи реалізації проєкту представлено на рисунку 3.6 [51] - [52].

Етап 1. Збір вимог та розробка технічного завдання (ТЗ):

- вивчення вихідної інформації;
- уточнення та формалізація вимог;
- визначення форматів даних;
- опис API для інтеграцій [51].

ТЗ у створенні вебпрограми – найважливіша частина, оскільки всі роботи будуть орієнтуватися на інформацію, вказану в ньому. Технічне завдання може змінюватись та доповнюватись від версії до версії [52].

ТЗ створюється з урахуванням того, що проєкт надалі матиме продовження та постійно модернізуватиметься, що веде до наступних правил проектування:

- проєкт описується частинами з чітким визначенням кінцевого працюючого продукту;
- у процесі покрокового проектування чітко вибудовується логіка впливу нових функцій на конверсію сайту, що зумовлює розуміння розвитку порталу.
- обмежень немає [52].

Етап 2. Прототипування:

- створення інтуїтивно зрозумілого та зручного інтерфейсу;
- проектування та розробка прототипу;
- Usability-тестування прототипу [51].

Для кожної версії створюються прототипи, що віддаються на тестування фокус-групі для максимального об'єктного аналізу [52].



Рисунок 3.6 – Етапи реалізації проєкту

Етап 3. Створення дизайну: розробка повнокольорового дизайну відповідно до розробленої структури та логіки, а також стильових побажань Замовника (UX/UI дизайн) [51].

Дизайн проєкту розробляється дизайнерами та затверджується замовником. До дизайну входить створення візуальної складової всіх елементів проєкту [52].

Етап 4. Верстка та розробка:

- верстка інтерфейсу вебзастосунка/сайту (frontend);
- розробка серверної частини (backend);
- інтеграція із суміжними системами (при необхідності) [51].

Етап 4.1. Backend – окремо проєктована частина сайту, яка включатиме максимальну автоматизацію роботи над проєктом в рамках поставленого завдання:

- архітектура бази даних та побудова взаємозв'язків;
- плановане виконання задач з завантаження та вивантаження даних;
- повідомлення та попередження про невиконання функцій вебзастосунка [52].

Етап 4.2. FrontEnd-верстка, js и controler. Верстка проєкту – це перетворення дизайну на працюючі вебелементи, кодинг формату дизайну сайту на формат вебзастосунку. JS дозволяє отримувати динамічно всю необхідну інформацію та звести до мінімуму час очікування. Controller на сервері дозволяє отримувати за захищеними каналами інформацію в потрібному для користувача вигляді [52].

Етап 5. Тестування: перевірка функціонування вебзастосунка/сайту, виправлення виявлених помилок, оптимізація [51].

Цей етап є одним з найбільш відповідальних, оскільки тестуються всі функції проєкту, які в ньому передбачені з різними умовами. Тестування проходить у два етапи: Альфа та Бета. На першому етапі створюється список

усіх необхідних доопрацювань, на другому етапі все ще перевіряється перед запуском проєкту [52].

Етап 6. Документування: розробка комплексу проєктної та експлуатаційної документації [51].

Етап 7. Запуск та маркетинг: передбачає розробку маркетингової стратегії розвитку [52].

3.5 Висновки до розділу 3

В даному розділі було розглянуто функціональні та нефункціональні вимоги до вебзастосунку. Розроблено архітектуру системи моніторингу, що містить клієнтську (для пацієнтів) та адміністративну (для лікарів) частини.

Також проаналізовано та обрано інструментарій розробки (мови програмування та середовище розробки).

Визначено етапи реалізації проєкту та їх особливості.

4 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Алгоритми функціонування системи

Схема алгоритму підключення до маршрутизаторів мережі показана на рисунку 4.1.

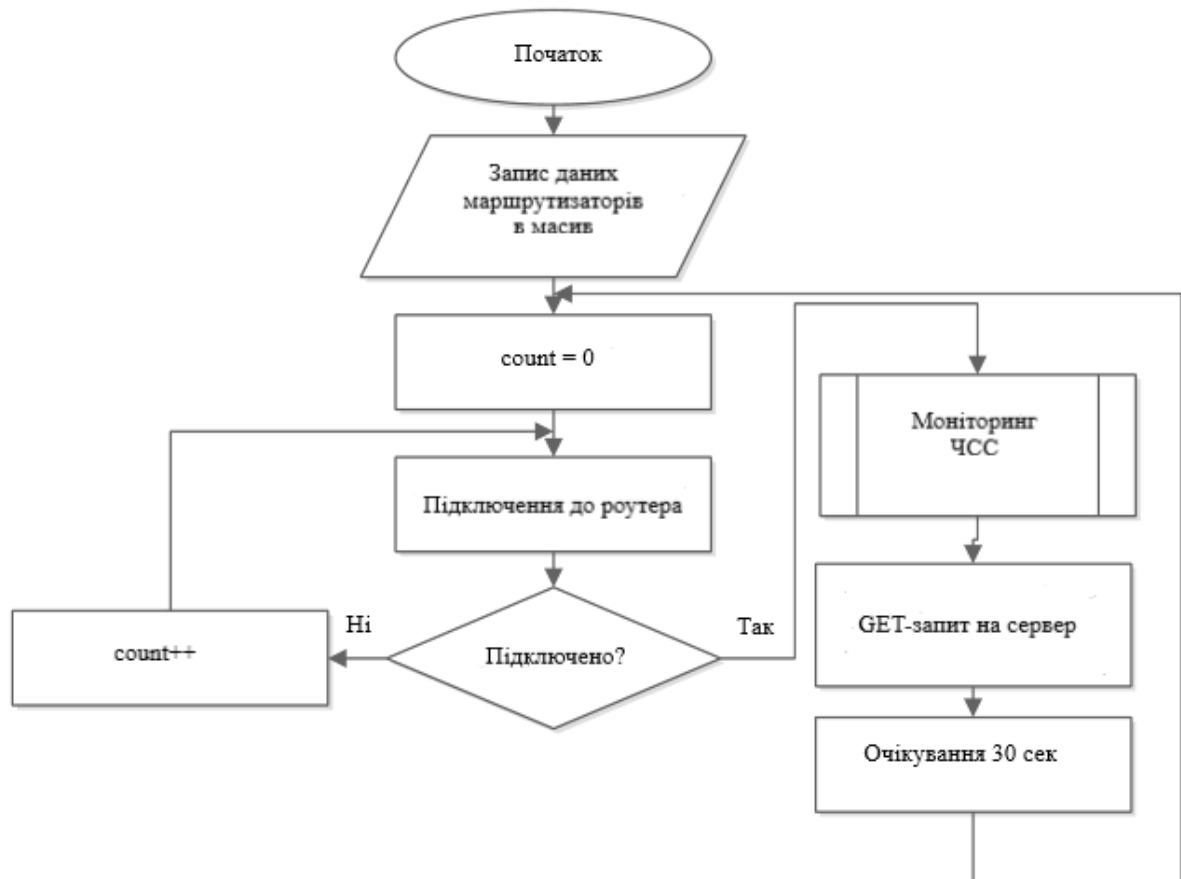


Рисунок 4.1 - Схема алгоритму підключення до маршрутизаторів мережі

Спочатку дані маршрутизаторів (їх логіни та паролі) записуються в масив пристрою.

Далі, в алгоритмі є певний лічильник (count), який у початковій позиції дорівнює 0 (count = 0).

Далі відбувається підключення до роутера, який знаходиться в радіусі пристрою, з даними, які беруться з масиву з індексом нашого лічильника (array [count]).

Якщо дані не підходять, лічильник збільшується на 1 і процедура повторюється доти, доки не буде вибрано дані, необхідні для підключення до маршрутизатора.

Після підключення відбувається моніторинг ЧСС, де зчитуються дані з носимого пристрою пацієнта.

Далі зібрані дані відправляються на сервер за допомогою запиту GET.

Після цього відбувається очікування 30 сек, щоб не перевантажувати сервер і процедура повторюється заново. Розмір вимірюваних даних становить 64 байти, що дозволяє уникнути навантаження на сервер.

Схема алгоритму моніторингу ЧСС показана на рисунку 4.2.

Спочатку вказуються значення за замовчуванням для оцінки сигналу від датчика ЧСС (`First_beat = true`, `Second_beat = true`, `array`).

Потім перевіряється умова перевищення сигналом критичних точок. У разі відсутності сигналу або його критичного значення (шуму) цей алгоритм завершує свою роботу.

Якщо сигнал пройшов перевірку стану, сигнал перевіряється, чи є він першим сигналом чи ні. Якщо це перший сигнал, то алгоритм завершує свою роботу.

Після цього зчитується наступний сигнал, який не є першим і передається далі за алгоритмом.

Далі значення сигналу додається до масиву, і воно повертається до зчитування наступного сигналу.

Після досягнення масивом 10 значень, які були виміряні, обчислюється середнє арифметичне.

Далі для цього середнього розраховується кількість ударів в хвилину. Результат записується для подальшої відправки на сервер.

Розробка програмного забезпечення для Wi-Fi модуля виконувалася в середовищі Arduino IDE на мові програмування C.

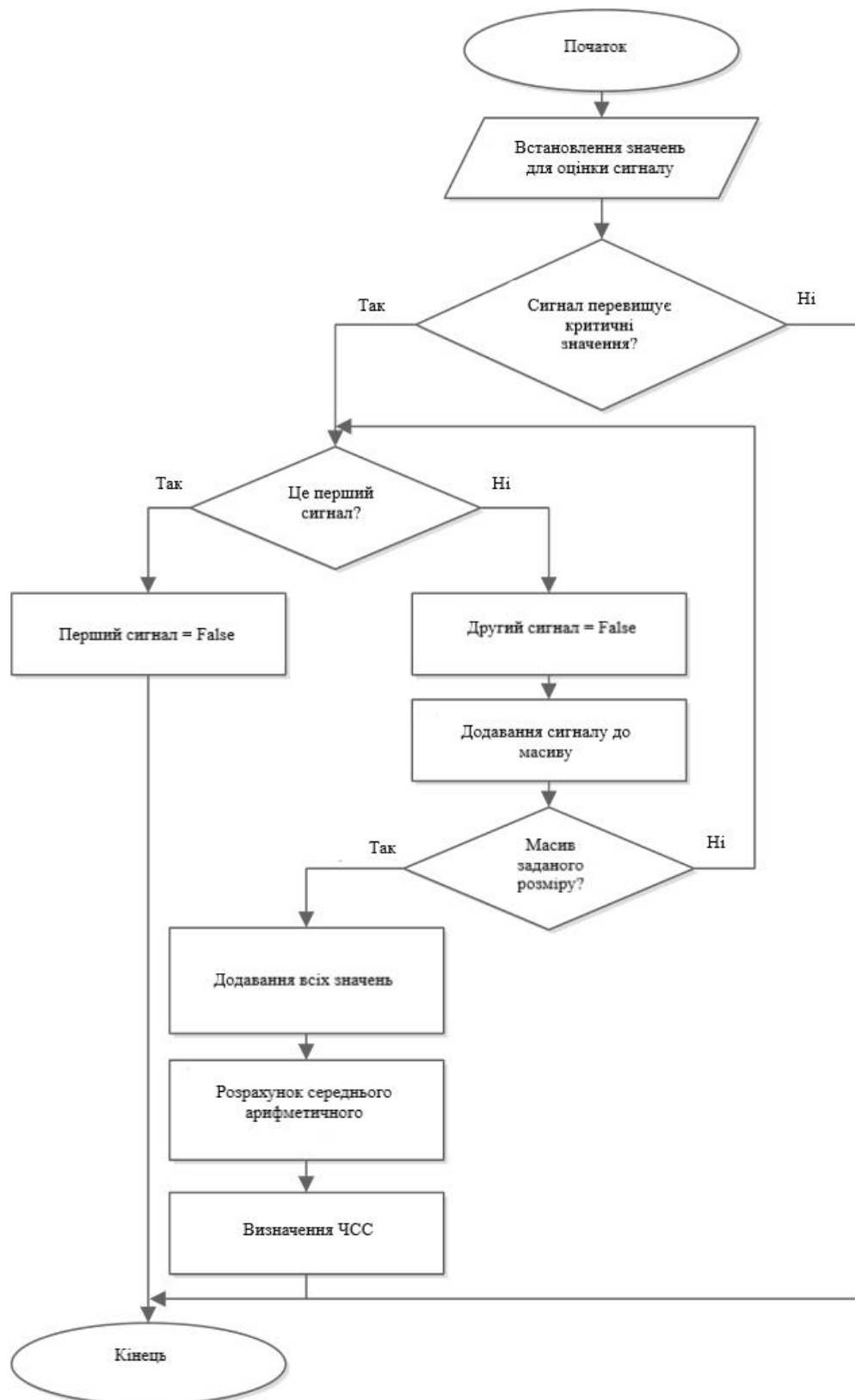


Рисунок 4.2 - Схема алгоритму моніторингу ЧСС

4.2 Програмна реалізація вебзастосунку

За основу інтерфейсу програмного продукту було прийнято інтуїтивно-зрозумілий дизайн оформлення: розроблено основну сторінку, з якої при роботі з вебзастосунком викликаються інші сторінки. Дизайн є досить зрозумілим для будь-якого користувача і не викликає жодних проблем у використанні, а отже є зручним для пацієнтів та лікарів.

Опис основних інтерфейсних модулів вебзастосунку наведено в таблиці 4.1.

Таблиця 4.1 – Опис основних модулів вебзастосунку

Об'єкт	Опис
index.html	Оформлення головної сторінки
about.html	Оформлення сторінки «ABOUT»
admin-panel.html	Оформлення сторінки моніторингу показників
login.html	Оформлення сторінки авторизації
custom.js	Головні функції вебзастосунку
main.js	Логіка функціоналу вебзастосунку на стороні пацієнта
swiper.js	Логіка функціоналу вебзастосунку на стороні лікаря

Фізичну структуру інтерфейсного модуля можливо переглянути на рисунку 4.3, текст програми для модулів index.html, login.html, admin-panel.html – в Додатку А.

В розробленому вебзастосунку для зручного введення даних використовувалися такі елементи як input, TextArea та Button:

- Input – це елемент для зручного введення невеликого тексту;
- TextArea – це елемент для зручного введення великого за обсягом тексту;
- Button – це елемент для зручного підтвердження та відправлення введених даних.

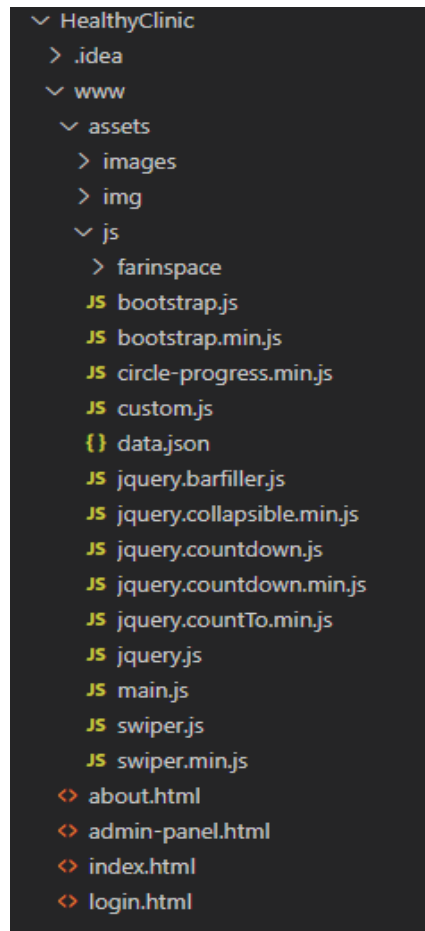


Рисунок 4.3 – Вигляд фізичної структури інтерфейсного модуля

Всі дані, що були введені, проходять перевірку на коректність введених даних та потрапляють до бази даних, якщо дані введено вірно. А якщо ж дані були введені неправильно, то виводиться інформація, в яких саме полях невірно введена інформація.

Обробка даних необхідна для зручної динамічної роботи сайту за допомогою сторінок вебзастосунку та даних.

В вебзастосунку обробка всіх даних відбувається в модулях: «main.js», «custom.js» та в «swiper.js». В цих модулях виконується обробка наступної інформації: додавання нових пацієнтів, редагування пацієнтів та видалення пацієнтів, що одужали, підписка на розсилання листів пацієнтам, авторизація лікарів, перевірка на правильність заповнення полів.

Наприклад, код одного з модулів вебзастосунку - це перевірка введених даних при додаванні нового пацієнта:

```

    if( name != "" && surname != "" && idBracelet != "" && numChamber
    != ""){
        $("#list-
patients").append("<tr id="+idBracelet+"><td>"+idBracelet+"</td><td>"+
name+"</td><td>"+surname+"</td><td>"+numChamber+"</td><td id='pulsus'+
idBracelet+"'>87</td><td><a href='#' class='delete'>x</a></td></tr>");
        $(".tbl-with-patients").css("display", "block");
        PopUpHide();

        $("#form-patients").trigger('reset');
        document.getElementById('form-patients').reset();
    } else{
        alert('Enter correct data!');
    }
}

```

Зовнішніми даними для вебзастосунку є дані, які були введені лікарем з клавіатури, а також дані з носимих пристроїв пацієнтів. Приклади зовнішніх даних для вебзастосунку:

- лікар авторизується на вебсторінці та вводить логін та пароль;
- лікар реєструє нового пацієнта та водить дані для реєстрації, що є зовнішніми даними.

Щоб створити динамічний сайт, потрібно використовувати програми, що імітують вебсервер. Це дозволяє оцінити, чи коректно відображається сайт у всіх поширених браузерях, і внести правки, якщо потрібно. Порівняльну характеристику найбільш популярних локальних серверів OpenServer та Denwer наведено в таблиці 4.2 [53] - [54].

У плані функціональності Open Server переважає Denwer. Також, у якості переваг Open Server можна виділити можливість переключення PHP/MySQL-модулів, зручність налаштування та управління компонентами, підтримку профілю налаштувань, приємний та продуманий інтерфейс [54].

Таблиця 4.2 – Порівняння популярних локальних серверів [54]

	OpenServer	Denwer
Розмір дистрибутива	413 Мб	8,2Мб
Версія PHP	Декілька на вибір	5.3.13
Версія MySQL	Декілька на вибір	5.5
Графічний інтерфейс	+	-
Багатомовність керуючої програми	+	-
Пакет додаткових програм	+	-
Підтримка профілів налаштувань та віртуальних піддоменів	+	-
Візуальні теми оформлення	+	-
Докладна документація	+	-
Оновлення	Кожні декілька місяців	Один раз за декілька років

При розробці вебзастосунку приділено увагу розробці інтерфейсу, оскільки інтерфейс користувача – це компоненти та елементи програмного продукту, які суттєво впливають на взаємодію користувача та програмного забезпечення [55].

В процесі розробки було використано кілька правил для розробки інтерфейсу, які дозволяють інтерфейсу вебзастосунку бути зрозумілим для лікаря та пацієнта.

Чим простіше тим і краще. Це правило не означає те, що в інтерфейсі лише трохи тексту та контактна інформація. Але на сайті не міститься зайвої інформації та шрифт такий, що легко читається. Графічні елементи швидко завантажуються, чіткі та виконані в одному стилі з оформленням сторінок вебзастосунку. В вебзастосунку не використовувались різноманітні звукові ефекти та анімація, адже це відволікає увагу користувачів, та призводить до збільшення часу завантаження сторінки.

Людина переглядає сторінки зверху до низу. Більше уваги зосереджується на верхній лівій частині сайту. Тому правильніше було розмістити найважливішу інформацію зверху, а саме: логотип, назву, контактні дані. Верхня частина сторінки є простою, це зроблено для того щоб відвідувачі вебзастосунку легше запам'ятовували інформацію, що там знаходиться.

Також у вебзастосунку є інформативні розділи для лікарів та пацієнтів.

Система управління вебзастосунку є невидимою для звичайних клієнтів. Вхід до адміністративної частини вебзастосунку може здійснити лише лікарем після того, як він введе правильний логін та пароль.

Лікар має можливість повного керування вмісту даними пацієнтів: переглядати, додавати, редагувати та видаляти будь-яку інформацію про пацієнтів та переглядати життєво важливі дані пацієнтів.

Інтерфейс для користувача простий та достатньо інформативний, щоб користувач потрапляючи до вебзастосунку отримував необхідну та чітку інформацію.

4.3 Висновки до розділу 4

В розділі було розглянуто узагальнені алгоритми підключення до мережі та моніторингу ЧСС на основі даних з носимих пристроїв пацієнтів.

Описано основні модулі розробленого вебзастосунку для моніторингу показників та структуру інтерфейсного модуля.

Також було представлено засоби, які були потрібні для розробки вебзастосунку, зокрема локальний сервер OpenServer, а також розглянуто особливості створення інтерфейсу.

5 КЕРІВНИЦТВО ЛІКАРЯ

5.1 Призначення й умови застосування програми

Вебзастосунок призначений для моніторингу лікарями показників фізичного стану пацієнтів, що використовують носимі пристрої на основі Wi-Fi модуля WeMos D1 та датчика Pulse Sensor.

Для зручної розробки вебзастосунку рекомендується мати ПК з такими характеристиками:

- процесор Intel (R) Core (TM) i5-3230M CPU 2.60GHz;
- оперативна пам'ять 8.00 ГБ (доступно 7,86);
- відеокарта NVIDIA GeForce GT 645M;
- 15 дюймовий монітор, клавіатура та маніпулятор «миша».

На ПК мають бути встановлені інтернет-браузер Google Chrome, операційна система Windows 10.

5.2 Характеристики програми

Вебзастосунок надає наступні функціональні можливості:

- відображення показників фізичного стану кожного пацієнта, що отримуються за допомогою носимого пристрою;
- можливість додавання нових пацієнтів шляхом заповнення форми, що містить поля для введення імені та прізвища, дентифікатора носимого пристрою, номера палати пацієнта.
- авторизація та надання доступу до адміністративної панелі лише лікарям.

5.3 Звернення до програми

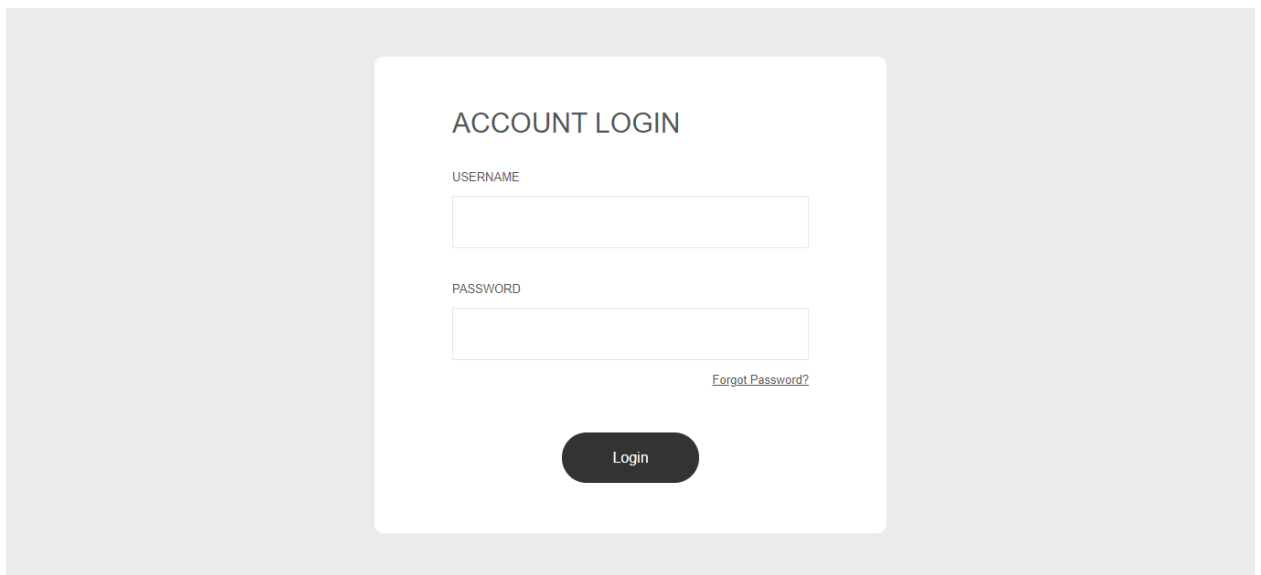
Для запуску програмного продукту необхідно мати встановлений Open Server, відкрити браузер та набрати в адресному рядку адресу `http://HealthyClinic/www/index.html` та натиснути клавішу «Enter».

Натиснувши на посилання «LOGIN» лівою кнопкою миші, буде перенаправлено на сторінку для авторизації лікаря. Сторінка «LOGIN» містить поля логіну та паролю, посилання «Forgot Password?» та кнопку «Login». Для авторизація лікаря в вебзастосунку необхідно вести коректний логін і пароль, та натиснути кнопку «Login».

При натисканні кнопки «Login» відбувається перевірка правильності введених даних та при правильно введених даних лікаря буде перенаправлено на головну адміністративну панель. Якщо ж дані були введені не коретно, то про це з'явиться повідомлення.

В тому разі, якщо лікар забув пароль, можливо натиснути на посілання «Forgot Password?» для його відновлення.

Сторінка авторизації лікаря представлена на рисунку 5.1.



ACCOUNT LOGIN

USERNAME

PASSWORD

[Forgot Password?](#)

Login

Рисунок 5.1 – Сторінка авторизації лікаря

При успішній авторизації, лікар потрапляє на сторінку моніторингу показників фізичного стану пацієнтів. На цій сторінці лікар може додати нового пацієнта, видалити такого, що одужав, або ж здійснювати перегляд за показниками фізичного стану своїх пацієнтів в реальному часі.

Вигляд сторінки моніторингу показників фізичного стану пацієнтів можливо переглянути на рисунку 5.2.

MEDICS PRO
We care for you

HOME ADD PATIENT ABOUT BLOG CONTACT

REGISTERED PATIENTS

Id bracelet	Name	Surname	Nº of palat	Pulsus
1	Olia	Simonenko	5	84 x
2	Vasya	Popovich	4	86 x
3	Mary	Ivanova	4	77 x
4	Ivan	Petrov	6	73 x
5	Mike	Larson	2	71 x

Age	Target HR Zone 50-85%	Average Maximum Heart Rate, 100%
20 years	100-170 beats per minute	200 beats per minute
30 years	95-162 beats per minute	190 beats per minute
35 years	93-157 beats per minute	185 beats per minute
40 years	90-153 beats per minute	180 beats per minute
45 years	88-149 beats per minute	175 beats per minute
50 years	85-145 beats per minute	170 beats per minute
55 years	83-140 beats per minute	165 beats per minute
60 years	80-136 beats per minute	160 beats per minute
65 years	78-132 beats per minute	155 beats per minute
70 years	75-128 beats per minute	150 beats per minute

Sign up for a health diagnosis

Why is heart rate monitoring so important? Maintaining and improving heart health is the most common reason to start training. Fortunately, this goal is easily achievable. The heart is the most important muscle in our body. While at rest, our heart works quite calmly, during exercise, the body's demands increase, and the heart is forced to change to provide working muscles with oxygen. The longer and more regularly we train, the more prepared our heart becomes. By measuring heart rate, we get more information about our heart condition. Heart rate tells you how hard we train, spend energy, and even evaluate our current emotional background.

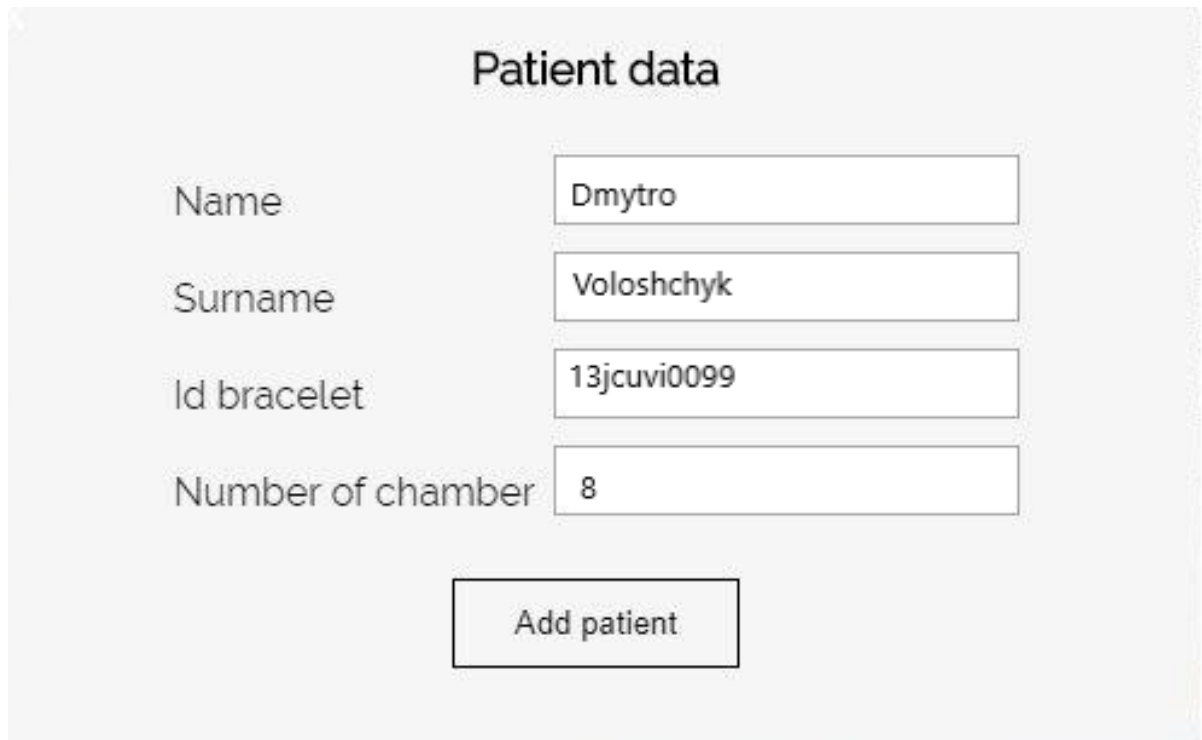
About Us
Services
About
Blog
Contact

Get in Touch
healthy-clinic@gmail.com
+380-66-87-30-329

Keep up with us Online
f t p

Рисунок 5.2 – Сторінка моніторингу показників фізичного стану пацієнтів

Натиснувши на посилання «ADD PATIENT», впливає форма для додавання нового пацієнта, що представлена на рисунку 5.3.



The image shows a web form titled "Patient data". It has four input fields arranged vertically. The first field is labeled "Name" and contains the text "Dmytro". The second field is labeled "Surname" and contains "Voloshchyk". The third field is labeled "Id bracelet" and contains "13jcuvi0099". The fourth field is labeled "Number of chamber" and contains "8". Below these fields is a rectangular button labeled "Add patient".

Рисунок 5.3 – Форма додавання пацієнта

Для того щоб додати успішно нового пацієнта необхідно коректно заповнити всі поля, а саме: «Name», «Surname», «Id bracelet», «Number of chamber» та натиснути кнопку «Add patient».

Після додавання нового клієнта, він відображається в списку всіх пацієнтів, де міститься Id bracelet, Name, Surname, № of palat, Pulsus всіх доданих пацієнтів (див. рис. 5.2).

Для видалення непотрбного пацієнта необхідна натиснути на червоний хрестик, який знаходиться напроти кожного пацієнта.

Також на сторінці моніторингу для зручності відображається таблиця з довідковими даними про вікові категорії та типові значення нормального пульсу та тиску, що повинні бути у пацієнтів.

5.4 Вхідні та вихідні дані

Вхідними даними до програми є дані, що надходять з носимих пристроїв пацієнтів, а саме ЧСС. Вихідними даними є збережена інформація про показники фізичного стану пацієнта за певний проміжок часу.

5.5 Повідомлення

У випадку неправильної авторизації лікаря, програма видасть відповідне повідомлення.

У випадку некоректного заповнення полів при реєстрації нового пацієнта, або наявності незаповнених полів, програма видасть відповідне повідомлення.

6 КЕРІВНИЦТВО ПАЦІЄНТА

6.1 Призначення програми

Програма призначена для моніторингу показників фізичного стану пацієнтів (ЧСС).

6.2 Умови виконання програми

Для зручної розробки вебзастосунку рекомендується мати ПК з такими характеристиками:

- процесор Intel (R) Core (TM) i5-3230M CPU 2.60GHz;
- оперативна пам'ять 8.00 ГБ (доступно 7,86);
- відеокарта NVIDIA GeForce GT 645M;
- 15 дюймовий монітор, клавіатура та маніпулятор «миша».

На ПК мають бути встановлені інтернет-браузер Google Chrome, операційна система Windows 10.

6.3 Робота з програмою

Даний програмний продукт не потребує інсталяції. Для роботи з вебзастосунком необхідно лише скопіювати папку з файлами розробленого вебзастосунка до серверу.

Для запуску програмного продукту необхідно мати встановлений Open Server, відкрити браузер та набрати в адресному рядку адресу <http://HealthyClinic/www/index.html> та натиснути клавішу «Enter», після чого потрапляємо до головної сторінки вебзастосунку (рис. 6.1).

Головна сторінка містить: слайдер, інформативні блоки, логотип, посилання на вкладку «HOME», посилання на сторінку «ABOUT», посилання на вкладку «BLOG», посилання на вкладку «CONTACT», посилання на

сторінку «LOGIN». Також внизу головної сторінки знаходиться «підвал» сайту з контактною інформацією.



Рисунок 6.1 – Головна сторінка вебзастосунку

Натиснувши на посилання «HOME» лівою кнопкою миші, буде перенаправлено на вкладку з слайдером та основною інформацією.

Натиснувши на посилання «ABOUT» лівою кнопкою миші, буде перенаправлено на сторінку, яка містить основну інформацію про клініку, її працівників. Також на сторінці «ABOUT» можливо підписатися на розсилання листів. Зовнішній вигляд сторінки «ABOUT» можливо переглянути на рисунку 6.2.

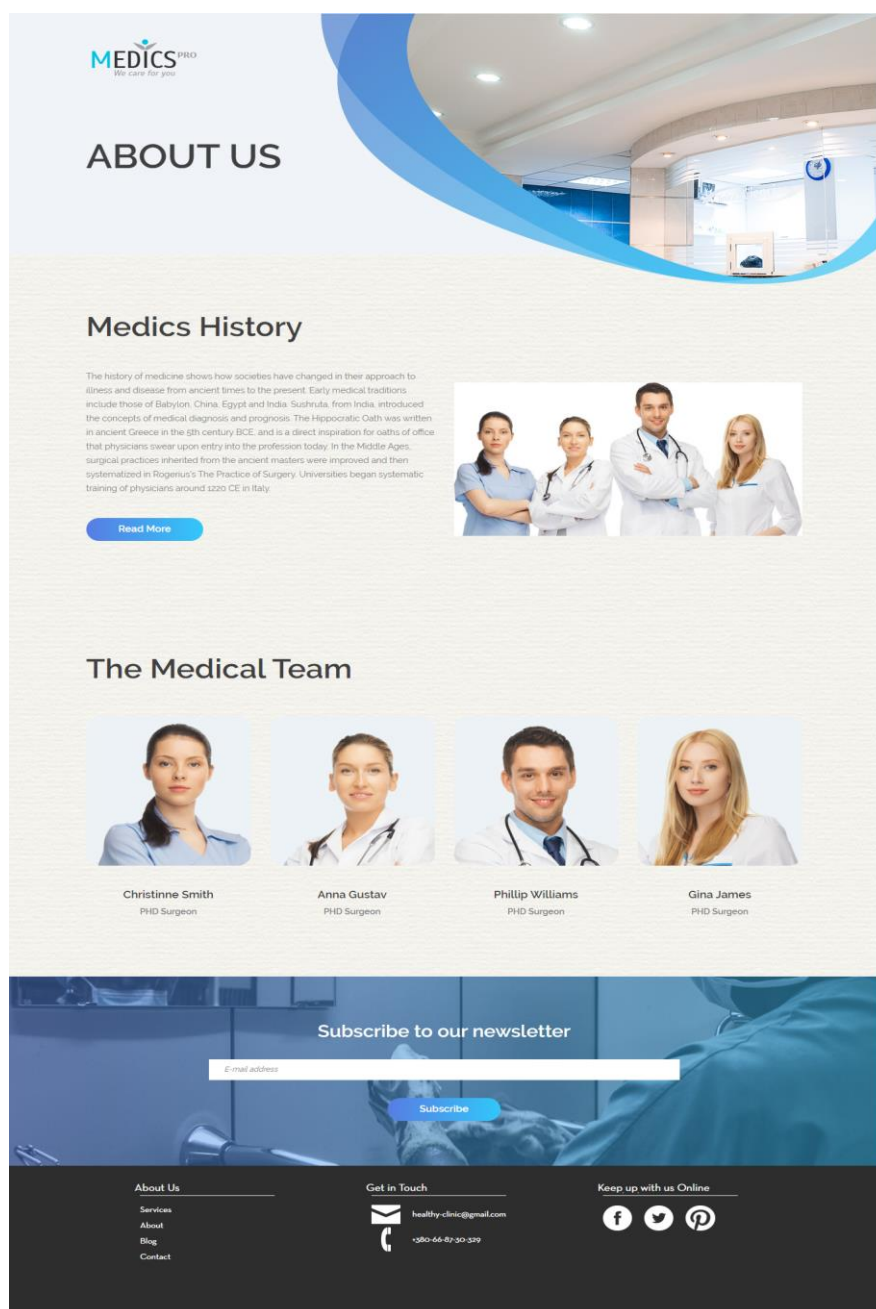


Рисунок 6.2 – Сторінка «ABOUT»

Натиснувши на посилання «BLOG» лівою кнопкою миші, буде перенаправлено на блок з інформаційним характером.

Натиснувши на посилання «CONTACT» лівою кнопкою миші, буде перенаправлено на блок з контактною інформацією про лікарню.

ВИСНОВКИ

У даній дипломній кваліфікаційній роботі було розроблено систему моніторингу показників фізичного стану пацієнтів на основі даних, які отримуються з носимого пристрою, закріпленого на руці пацієнта. Розроблений вебзастосунок може використовуватися в медичних закладах, в яких лікарі мають постійно слідкувати за станом здоров'я пацієнтів. Його застосування полегшує роботу персоналу щодо догляду за пацієнтами та надання своєчасної допомоги.

Вебзастосунок створений за допомогою мови гіпертекстової розмітки HTML, каскадних таблиць стилів CSS, об'єктно-орієнтованої мови програмування Java Script.

Розроблений застосунок надає такі функції: авторизація лікаря, додавання нових пацієнтів чи видалення таких, що одужали, перегляд повного списку пацієнтів з носимими пристроями та показників стану їх здоров'я.

Вебзастосунок може працювати під будь-якою ОС, при наявності доступу до інтернету та наявності інтернет-браузера.

Всі поставлені задачі були повністю вирішені, а саме:

- аналіз та вибір апаратних засобів для вимірювання показників фізичного стану пацієнтів та їх передачі на сервер;
- аналіз вимог до розроблюваного програмного забезпечення та вибір інструментарію розробки;
- розробка вебзастосунку та перевірка його роботи на локальному сервері;
- розробка програмної документації.

Працездатність вебзастосунку підтверджена в процесі тестування, а його застосування забезпечує в реальному часі інформацію лікарям, що підвищує оперативність надання медичної допомоги.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Понятие о здоровье, основные показатели здоровья [Электронный ресурс]. – Режим доступа: <https://psyera.ru/ponyatie-o-zdorove-osnovnyie-pokazateli-zdorovya-1479.htm>.
2. Двенадцать показателей организма, которые надо знать каждому [Электронный ресурс]. – Режим доступа: <https://weekend.rambler.ru/items/36266235-dvenadtsat-pokazateley-organizma-kotorye-nado-znat-kazhdomu/>.
3. Физиологические основы здоровья человека: учеб.-метод. пособие / [П.Н. Смирнов, Н.В. Ефанова, Л.М. Осина и др.]. – Новосибирск: ИЦ «Золотой колос», 2016. – 228 с.
4. Параметры здоровья [Электронный ресурс]. – Режим доступа: https://studopedia.su/4_20400_parametri-zdorovya.html.
5. Что такое зоны ЧСС, как их узнать и зачем это нужно [Электронный ресурс]. – Режим доступа: <https://nogibogi.com/pulsovyie-zony/>.
6. Артеріальний тиск і артеріальна гіпертензія [Електронний ресурс]. – Режим доступу: <http://clinic.gov.ua/?p=5798>.
7. Інформаційні технології в медицині [Електронний ресурс]. – Режим доступу: <http://surl.li/btpsw>.
8. Інноваційні технології в медицині [Електронний ресурс]. – Режим доступу: <https://www.slideshare.net/innagrabobska/ss-12937918>
9. Клініка Борис [Електронний ресурс]. – Режим доступу: <http://boris.kiev.ua/>.
10. Охматдит [Електронний ресурс]. – Режим доступу: <https://ohmatdyt.com.ua/>.
11. Клініка Святого Миколая [Електронний ресурс]. – Режим доступу: <https://ksm-clinic.com.ua/>.
12. HELSI [Електронний ресурс]. – Режим доступу: <https://helsi.me/>.

13. Фитнес-браслет [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Фитнес-трекер>.
14. Пульсометр [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Пульсометр>.
15. Топ-3 пульсометров для сердечников [Электронный ресурс]. – Режим доступа: https://zn.ua/novosti-kompaniy/top-3-pulsometrov-dlya-serdechnikov-310342_.html.
16. BOXYM medical fingertip pulse oximeter blood oxygen heart rate monitor digital LED oximetro health monitors oximetro de pulso [Electronic source]. – Access mode: <https://ziloqa.in/product/boxym-medical-fingertip-pulse-oximeter-blood-oxygen-heart-rate-monitor-digital-led-oximetro-health-monitors-oximetro-de-pulso/>.
17. A guide to finger pulse oximeters for 2022 [Electronic source]. – Access mode: <https://www.healthline.com/health/finger-pulse-oximeter>.
18. Что такое смарт-часы, для чего они нужны и что умеют делать [Электронный ресурс]. – Режим доступа: <https://smartchasy.com/smart-chasy/chto-takoe-smart-chasy-dlja-chego-oni-nuzhny-i-chto-umejut-delat/>.
19. Месяц с Apple Watch Series 3: исчерпывающий обзор [Электронный ресурс]. – Режим доступа: <https://lifehacker.ru/apple-watch-series-3-review/>.
20. Aminian, M. A hospital healthcare monitoring system using wireless sensor networks / M. Aminian, H.R. Naji // Journal of Health and Medical Information. – 2013. – Vol. 4, Issue 2. – P.1-6.
21. Kakria, P. A real-time health monitoring system for remote cardiac patients using smartphone and wearable sensors / P. Kakria, N.K. Tripathi, P. Kitipawang // International Journal of Telemedicine and Applications. – 2015. - Vol. 11. – P.1-11 p.

22. Novii wireless patch system [Electronic source]. – Access mode:<http://www.monicahealthcare.com/products/labour-and-delivery/monica-novii-wireless-patch-system>
23. IReHMo: an efficient IoT-based remote health monitoring system for smart regions / [N. M. Khoi, S. Saguna, K. Mitra et.al] // E-health Networking, Application&Services (HealthCom): 17th international conference, Boston, MA, USA, 14–17 October 2015: proceedings. – 2015.- P. 563-568.
24. Parihar, V.R. Heartbeat and temperature monitoring system for remote patients using Arduino/ V. R. Parihar, A. Y. Tonge, P. D. Ganorkar // International Journal of Advanced Engineering Research and Science. – 2017. – Vol. 4, Issue 5. – P. 55-58.
25. Das, C. K. A wireless heartbeat and temperature monitoring system for remote patients / C. K. Das, M. W. Alam, M. I. Hoque // Mechanical engineering and renewable energy (ICMERE): International conference, Chittagong, Bangladesh, 1-3 May 2014: proceedings. – 2014. – P. 1-7.
26. Secured smart healthcare monitoring system based on IoT/ [A. Divya, K.Keerthana, N.Kiruthikanjali et.al] // Asian Journal of Applied Science and Technology (AJAST). – 2017. – Vol.1, Issue 2. – P. 53-56.
27. Alam, M.W. A heartbeat and temperature measuring system for remote health monitoring using wireless body area network”/ M. W. Alam, T. Sultana, M. S. Alam // International Journal of Bio-Science and Bio-Technology. – 2016. – Vol. 8, Issue 1. – P.171-190.
28. Remote heart rate monitoring of the hospital patients / [A. Parkhomenko, Y. Presaizen, O. Gladkova et.al] // Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS): 10th IEEE International conference, Metz, France, 18-21 September 2019: proceedings. - 2019. – P. 991-996.

29. D1 MINI D1 Mini - ESP8266, v2.0 [Electronic source]. – Access mode: <https://www.reichelt.com/de/en/d1-mini-esp8266-v2-0-d1-mini-p253978.html>.
30. Pulse Sensor [Electronic source]. – Access mode: <https://www.sparkfun.com/products/11574>.
31. Датчик пульса, сердечного ритма PulseSensor [Электрон. ресурс]. – Режим доступа: <https://controller.in.ua/sensori-%E2%86%92-taktilnie>.
32. KY-039 – ИК датчик измерения пульса [Электрон. ресурс]. – Режим доступа: <https://voltiq.ru/shop/finger-measuring-heartbeat-sensor/>.
33. CJMCU-pulse sensor pulse heart rate sensor pulse sensor module rate detecting monitoring [Electronic resource]. – Access mode: <https://www.thanksbuyer.com/cjmcu-pulse-sensor-pulse-heart-rate-sensor-pulse-sensor-module-rate-detecting-monitoring-35055>.
34. Ермилов, С. Соединяем монитор частоты сердечных сокращений max30100 с Ардуино [Электрон. ресурс] / С. Ермилов. – Режим доступа: <https://arduinoplus.ru/arduino-max30100/>.
35. AD8232 – модуль для ЭКГ [Электрон. ресурс]. – Режим доступа: <https://controller.in.ua/ad8232---modul-dlja-jekg>.
36. PSL-iECG2 (Mini-size 2ch ECG sensor module with isolation) [Electronic resource]. – Access mode: https://physiolab.en.ec21.com/PSL-iECG2-Mini-size-2ch--9063127_9816937.html.
37. Створення структури документа HTML. Робота з основними тегами HTML [Електронний ресурс]. – Режим доступу: <https://bevgeny-vpu.blogspot.com/2016/02/26-html-html.html>
38. Муссиано, Ч. HTML & XHTML. Подробное руководство / Ч. Муссиано, Б. Кеннеди. – К.: Символ-Плюс, 2008. – 748с.
39. CSS — Что такое, история создания и преимущества [Электрон. ресурс]. – Режим доступа: <http://surl.li/bucni>.

40. Топ 10 языков программирования [Электрон. ресурс]. – Режим доступа: <http://surl.li/buibr>.
41. Дунаев В.В. Самоучитель JavaScript. 3-е изд. / В.В. Дунаев. – М.: Питер, 2008.– 400с.
42. Знакомство с jQuery [Электрон. ресурс]. – Режим доступа: <https://cyberpedia.su/9x131d9.html>.
43. Язык JavaScript [Электрон. ресурс]. – Режим доступа: <https://www.mindomo.com/ru/mindmap/web-4-0876948771c2407fb8c8d3cc96be0297>.
44. Введение в AJAX и COMET [Электрон. ресурс]. – Режим доступа: <https://learn.javascript.ru/ajax-intro>.
45. AJAX [Электрон. ресурс]. – Режим доступа: <https://astwellsoft.com/ru/blog/tehnology/ajax.html>.
46. Получение данных с сервера [Электрон. ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data.
47. 20 best IDE softwares for coding [Electronic resource]. – Access mode: <https://geekyhumans.com/de/20-best-ide/>.
48. PhpStorm [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PhpStorm>.
49. Visual Studio Code [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Visual_Studio_Code.
50. Формат JSON, метод toJSON [Электрон. ресурс]. – Режим доступа: <http://surl.li/bujlu>.
51. Этапы создания веб-приложений и сайтов [Электрон. ресурс]. – Режим доступа: <https://uncore.ru/blog/development/etapy-sozdaniya-web-prilozhenij-i-sajtov/>.
52. Профессиональная разработка веб-приложений [Электрон. ресурс]. – Режим доступа: <https://mandarin5.ru/razrabotka-web-prilozhenij/>.

53. Какой локальный сервер лучше: OPENSERVR или DENWER [Электрон. ресурс]. – Режим доступа: <https://nastroyvse.ru/programs/rating/chto-vybrat-openserver-ili-denwer.html>.

54. Устанавливаем CMS на локальный сервер: Denwer vs Open Server [Электрон. ресурс]. – Режим доступа: <https://www.imagecms.net/blog/veb-razrabotka/denwer-vs-open-server>

55. Маркотт И. Отзывчивый веб-дизайн / И. Маркотт. - М.: Манн, Иванов и Фербер, 2012. – 277 с.

Додаток А
Текст програми

A.1 Код index.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Healthy Clinic</title>
  <link rel="icon" href="assets/images/favicon.ico">
  <link href="https://fonts.googleapis.com/css?family=Raleway&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="assets/css/main.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.mi
n.js"></script>
  <script type="text/javascript" src="assets/js/farinspace/jquery.imgpreloa
d.min.js"></script>

</head>
<body>
  <header>
    <div class="centering">
      <a href="#">
        
      </a>
      <div class="header_right">
        <nav class="nav right">
          <ul>
            <li class="current">
              <a href="#">Home</a>
            </li>
            <li>
              <a href="about.html">About</a>
            </li>
            <li>
              <a href="#forBlog">Blog</a>
            </li>
            <li>
              <a href="#contact-us">Contact</a>
            </li>
            <li>
              <a href="login.html">Login</a>
            </li>
          </ul>
        </nav>
      </div>
    </div>
  </header>

  <section class="slider">
    <div class="marquee_container autoplay">
      <div class="marquee_photos"></div>

      <div class="marquee_nav"></div>
    </div>

    <div class="marquee_panels">
      <div class="marquee_panel">

```

```

        
    </div>
    <div class="marquee_panel">
        
    </div>
    <div class="marquee_panel">
        
    </div>
</div>
</section>

<section class="visitor_info centering indent">
    <h2>OUR TECHNOLOGICAL SERVICES</h2>

    <div class="about_us">
        
        <h4><a href="#">ADVANCED TECHNOLOGY</a></h4>
        <p>Use modern technology to monitor your health.</p>
    </div>
    <div class="about_us mailing_list">
        
        <h4><a href="#">HEALTHCARE SOLUTIONS</a></h4>
        <p>The service uses technologies for real-
time health diagnostics.</p>
    </div>
    <div class="about_us">
        
        <h4><a href="#">24/7 AVAILABILITY</a></h4>
        <p>You can use the service at any time of the day, from any d
evice.</p>
    </div>
</section>

<section id="contact-
us" class="diagnostic_record centering indent bg_white">
    <div class="our_photo">
        
    </div>
    <div class="info_for_record">
        <h3>Sign up for a health diagnosis</h3>
        <form>
            <div>
                <label for="name">Name</label>
                <input type="text" id="name">
            </div>
            <div>
                <label for="email">Email Address</label>
                <input type="email" id="email">
            </div>
        </form>
        <button type="button">Sign Up</button>
    </div>
</section>

```

```

<section id="forAbout" class="centering indent bg_white data_measurement"
>
    <div class="info_for_record">
        <h3>What is the pulse?</h3>
        <p>Pulse, rhythmic dilation of an artery generated by the opening
and closing of the aortic valve in the heart. A pulse can be felt by applying firm f
ingertip pressure to the skin at sites where the arteries travel near the skin's surf
ace; it is more evident when surrounding muscles are relaxed. Common pulse points inc
lude the carotid artery of the neck, the brachial artery inside the elbow, and the ra
dial artery in the wrist.</p>
        <p style="margin-
top: 20px;">You can measure your heart rate with us.</p>
    </div>
    <div class="output-data">
        <div id="data">
            <p class="count_pulse">85
            <span>
                <svg xml:space="preserve" viewBox="0 0 100 100" y="0"
x="0" xmlns="http://www.w3.org/2000/svg" id="圖層_1" version="1.1" style="height: 40
px; width: 40px; background: rgb(255, 255, 255);" width="128px" height="128px"><g cla
ss="ld1-scale" style="transform-
origin: 50% 50%; transform: scale(0.8);"><g class="ld1-ani"><g class="ld1-
layer"><g class="ld1-ani" style="transform: scale(0.91); transform-
origin: 50px 50px; animation: 1.11111s linear -
1.11111s infinite normal forwards running breath-4a686512-feed-44d2-b939-
3a02798e896d;"><path fill="#ed1e79" d="M90 34.613a21.48 21.48 0 0 0-.451-4.378c-
2.021-9.724-10.635-17.032-20.959-17.032-4.43 0-8.545 1.346-
11.96 3.65A21.505 21.505 0 0 0 50 24.019a21.505 21.505 0 0 0-6.63-
7.166 21.311 21.311 0 0 0-11.96-3.65c-10.324 0-18.938 7.308-
20.959 17.032A21.48 21.48 0 0 0 10 34.613c0 .328.022.651.037.975-.002.159-.012.315-
.012.474C10.025 58.179 50 86.797 50 86.797s39.975-28.618 39.975-50.734c0-.159-.01-
.316-.012-.474.015-.325.037-.648.037-
.976z" style="fill: rgb(237, 30, 121);"></path></g></g><metadata xmlns:d="https://loa
ding.io/stock/"><d:name>heart</d:name><d:tags>love,like,favorite,heart,people</d:tags
><d:license>by</d:license><d:slug>a7634f</d:slug></metadata></g></g></svg>
                BPM
            </span>
        </p>
        <p class="subtitleCount">Normal measurement pulse</p>
    </div>
    <div class="generalValue">
        <div class="moreInfo">
            <p>General</p>
            <p class="dateTime">14.12.2019, 14:00</p>
        </div>
        <div class="generalValuePulse">
            87 <span>BPM</span>
        </div>
    </div>
</div>
</section>

<section id="forBlog" class="diagnostic_record info_rate centering indent
bg_white">
    <div class="our_photo">
        
    </div>
    <div class="info_for_record">
        <h3>Sign up for a health diagnosis</h3>

```

<p>Why is heart rate monitoring so important?
 Maintaining and improving heart health is the most common reason to start training. Fortunately, this goal is easily achievable.

The heart is the most important muscle in our body. While at rest, our heart works quite calmly, during exercise, the body's demands increase, and the heart is forced to change to provide working muscles with oxygen. The longer and more regularly we train, the more prepared our heart becomes.

By measuring heart rate, we get more information about our heart condition. Heart rate tells you how hard we train, spend energy, and even evaluate our current emotional background.</p>

</div>
 </section>

<footer>

<div class="centering">
 <div class="info_company about_company">
 <h3>About Us</h3>

 Services

 About

 Blog

 Contact

 </div>

<div class="info_company connect">
 <h3>Get in Touch</h3>
 <div class="email">

healthy-clinic@gmail.com

</div>
 <div class="phone">

 +380-66-87-30-329

</div>
 </div>

<div class="info_company social">
 <h3>Keep up with us Online</h3>


```

        </a>
      </div>
    </div>
  </footer>

  <script type="text/javascript" src="assets/js/main.js"></script>
</body>
</html>

```

A.2 Код login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Login Clinic</title>
  <link rel="icon" href="assets/images/favicon.ico">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="assets/css/main-lg.css">
  <link rel="stylesheet" href="assets/css/util.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.mi
n.js"></script>

</head>
<body>

  <div class="limiter">
    <div class="container-login100">
      <div class="wrap-login100 p-l-85 p-r-85 p-t-55 p-b-55">
        <form class="login100-form validate-form flex-sb flex-w">
          <span class="login100-form-title p-b-32">
            Account Login
          </span>

          <span class="txt1 p-b-11">
            Username
          </span>
          <div class="wrap-input100 validate-input m-b-36" data-
validate = "Username is required">
            <input class="input100" type="text" name="username" i
d="login">

            <span class="focus-input100"></span>
          </div>

          <span class="txt1 p-b-11">
            Password
          </span>
          <div class="wrap-input100 validate-input m-b-12" data-
validate = "Password is required">
            <span class="btn-show-pass">
              <i class="fa fa-eye"></i>
            </span>
            <input class="input100" type="password" name="pass" i
d="password">

            <span class="focus-input100"></span>

```

```

        </div>

        <div class="flex-sb-m w-full p-b-48">
            <div class="contact100-form-checkbox">

                </div>

                <div>
                    <a href="#" class="txt3">
                        Forgot Password?
                    </a>
                </div>
            </div>

            <div class="container-login100-form-btn">
                <button class="login100-form-btn">
                    Login
                </button>
            </div>

        </form>
    </div>
</div>

<div id="dropDownSelect1"></div>

<script src="assets/js/main.js"></script>

</body>
</html>

```

A.3 Код admin-panel.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Admin Panel</title>
    <link rel="icon" href="assets/images/favicon.ico">
    <link href="https://fonts.googleapis.com/css?family=Raleway&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="assets/css/main.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.mi
n.js"></script>
    <script type="text/javascript" src="assets/js/farinspace/jquery.imgpreloa
d.min.js"></script>
</head>
<body>
    <header>
        <div class="centering">
            <a href="#">
                
            </a>

```

```

<div class="header_right">
  <nav class="nav right">
    <ul>
      <li class="current">
        <a href="#">Home</a>
      </li>
      <li>
        <a href="#" class="addPatient">Add patient</a>
      </li>
      <li>
        <a href="about.html">About</a>
      </li>
      <li>
        <a href="#">Blog</a>
      </li>
      <li>
        <a href="#">Contact</a>
      </li>
    </ul>
  </nav>
</div>
</div>
</header>

<section class="tbl-with-patients centering indent bg_white">
  <h2 class="titleTbl">Registered patients</h2>

  <table id="list-patients">
    <tr>
      <th>Id bracelet</th>
      <th>Name</th>
      <th>Surname</th>
      <th>№ of palat</th>
      <th>Pulsus</th>
      <th></th>
    </tr>
  </table>
</section>

<section class="imgPulsus centering indent bg_white">
  <div>
    
  </div>
</section>

<section class="diagnostic_record info_rate centering indent bg_white">
  <div class="info_for_record">
    <h3>Sign up for a health diagnosis</h3>
    <p>Why is heart rate monitoring so important?
      Maintaining and improving heart health is the most common rea
son to start training. Fortunately, this goal is easily achievable.

      The heart is the most important muscle in our body. While at
rest, our heart works quite calmly, during exercise, the body's demands increase, and
the heart is forced to change to provide working muscles with oxygen. The longer and
more regularly we train, the more prepared our heart becomes.

```

By measuring heart rate, we get more information about our heart condition. Heart rate tells you how hard we train, spend energy, and even evaluate our current emotional background.</p>

```

    </div>
    <div class="our_photo">
        
    </div>

</section>

<footer>
    <div class="centering">
        <div class="info_company about_company">
            <h3>About Us</h3>
            <ul>
                <li>
                    <a href="#">Services</a>
                </li>
                <li>
                    <a href="about.html">About</a>
                </li>
                <li>
                    <a href="#">Blog</a>
                </li>
                <li>
                    <a href="#">Contact</a>
                </li>
            </ul>
        </div>

        <div class="info_company connect">
            <h3>Get in Touch</h3>
            <div class="email">
                <a href="mailto:oliasimonenko1996@gmail.com" title="Send
message!">
                    healthy-clinic@gmail.com
                </a>
            </div>
            <div class="phone">
                <a href="tel:+380668730329" title="Call us by phone!">
                    +380-66-87-30-329
                </a>
            </div>
        </div>

        <div class="info_company social">
            <h3>Keep up with us Online</h3>
            <a href="#">
                
            </a>
            <a href="#">
                
            </a>
            <a href="#">
                
            </a>
        </div>
    </div>
</footer>

```

```

<div class="b-popup" id="popup1">
  <div class="b-popup-content">
    <a href="javascript:PopUpHide()">x</a>
    <h3>Patient data</h3>
    <div>
      <form id="form-patients">
        <div>
          <label for="name">Name</label>
          <input type="text" id="name">
        </div>
        <div>
          <label for="surname">Surname</label>
          <input type="text" id="surname">
        </div>
        <div>
          <label for="idBracelet">Id bracelet</label>
          <input type="number" id="idBracelet">
        </div>
        <div>
          <label for="numChamber">Number of chamber</label>
          <input type="number" id="numChamber">
        </div>
        <button class="addToList">Add patient</button>
      </form>
    </div>
  </div>
</div>

<script type="text/javascript" src="assets/js/main.js"></script>
</body>
</html>

```

Додаток Б
Слайди презентації

Національний університет «Запорізька політехніка»

Кафедра програмних засобів

Дипломна кваліфікаційна робота бакалавра
на тему:

Програмне забезпечення для віддаленого моніторингу показників фізичного стану пацієнтів

Виконав
студент групи КНТ-128
Керівник
к.т.н., доцент

Волощук Д.О.
Пархоменко А.В.

2022

Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

3

Мета та задачі роботи

Мета роботи: розробка вебзастосунку для віддаленого моніторингу показників фізичного стану здоров'я пацієнтів, що надасть інформацію лікарям в реальному часі та дозволить підвищити оперативність надання медичної допомоги.

Задачі роботи:

- аналіз та вибір апаратних засобів для вимірювання показників фізичного стану пацієнтів та їх передачі на сервер;
- аналіз вимог до розроблюваного програмного забезпечення та вибір інструментарію розробки;
- розробка вебзастосунку та перевірка його роботи на локальному сервері;
- розробка програмної документації.

Рисунок Б.3 – Слайд 3

4

Аналіз та вибір апаратних засобів

Порівняння модифікацій Wi-Fi модуля ESP8266

	Кількість GPIO виводів	Антенa	АЦП	Bluetooth	Вартість, \$
ESP-01	3	-	-	-	3
ESP-07	9	+	-	-	2.97
ESP-12F	12	+	1	-	3.04
ESP32	32	+	14	+	6.72

Носимий пристрій

Програмне забезпечення

Апаратне забезпечення

Модуль зчитування даних з датчика пульсу

Модуль зчитування даних з акселерометра

Модуль реалізації та перевірки підключення

Модуль відправки даних

Модуль виведення/видавання інформації

Модуль оповіщення

Механічна частина

Електронна частина

Корпус

Індикація

Кріплення

Ремінець

Дисплей

Головний МК

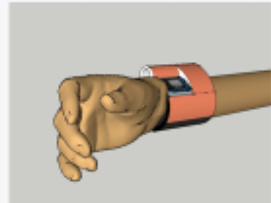
Акумулятор

Датчик ЧСС

Датчик акселерометра та гіроскопа



Pulse Sensor



Прототип носимого пристрою



WeMos D1 Mini ESP8266

Рисунок Б.4 – Слайд 4



Рисунок Б.5 – Слайд 5



Рисунок Б.6 – Слайд 6



Рисунок Б.7 – Слайд 7



Рисунок Б.8 – Слайд 8

Структура проекту

Об'єкт	Опис
index.html	Оформлення головної сторінки
about.html	Оформлення сторінки «ABOUT»
admin-panel.html	Оформлення сторінки моніторингу показників
login.html	Оформлення сторінки авторизації
custom.js	Головні функції вебзастосунку
main.js	Логіка функціоналу вебзастосунку на стороні пацієнта
swiper.js	Логіка функціоналу вебзастосунку на стороні лікаря

```
HealthyClinic
├── .idea
├── www
├── assets
│   ├── images
│   ├── img
│   └── js
│       ├── farinspace
│       ├── bootstrap.js
│       ├── bootstrap.min.js
│       ├── circle-progress.min.js
│       ├── custom.js
│       ├── data.json
│       ├── jquery.barfiller.js
│       ├── jquery.collapsible.min.js
│       ├── jquery.countdown.js
│       ├── jquery.countdown.min.js
│       ├── jquery.countTo.min.js
│       ├── jquery.js
│       ├── main.js
│       ├── swiper.js
│       └── swiper.min.js
├── about.html
├── admin-panel.html
├── index.html
└── login.html
```

Рисунок Б.9 – Слайд 9

Інтерфейс вебзастосунку

Головна сторінка вебзастосунку

Сторінка «ABOUT»

Рисунок Б.10 – Слайд 10

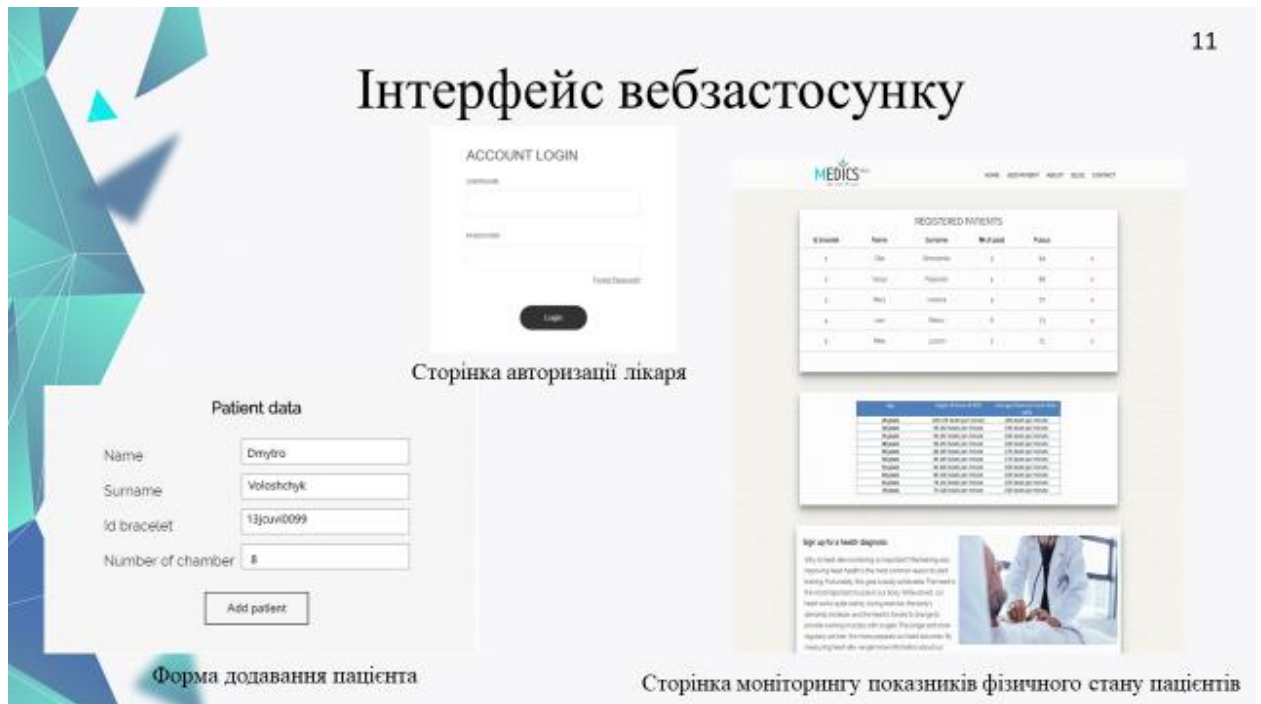


Рисунок Б.11 – Слайд 11

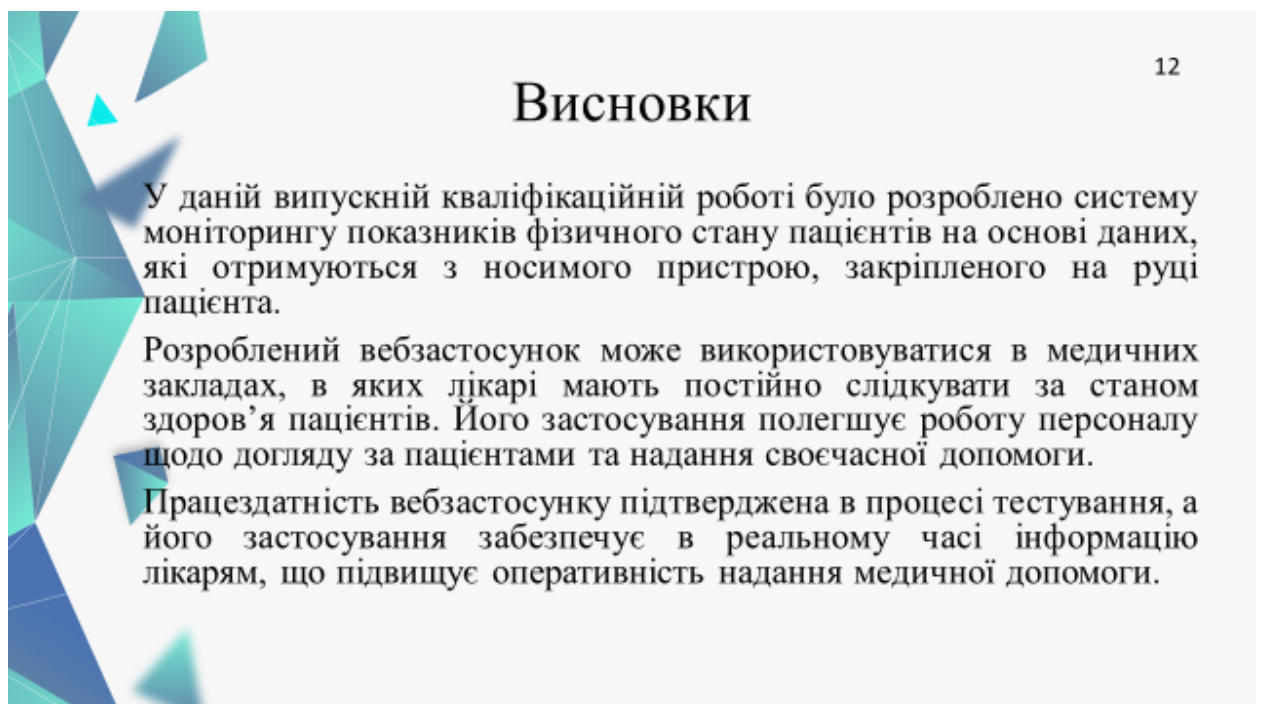


Рисунок Б.12 – Слайд 12



Рисунок Б.13 – Слайд 13