

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
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 ТРЕНУВАННЯ ОЧЕЙ
EYE TRAINING SOFTWARE

Виконав: студент(ка) 4 курсу, групи КНТ-227
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

Щербаков О.О.

(прізвище та ініціали)

Керівник Неласа Г.В.

(прізвище та ініціали)

Рецензент Гофман Є.О.

(прізвище та ініціали)

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

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Щербакова Олексія Олексійовича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для тренування очей
Eye Training Software

керівник проєкту (роботи) Неласа Ганна Вікторівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 24 травня 2021 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація та тестування програми. 5. Керівництво програміста.

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
1-5 Основна частина	Неласа Г.В., доцент		
Нормоконтролер	Андреєв М.О., асистент		

7. Дата видачі завдання “ 15 ” березня 2021 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 та дослідження предметної області	2 тиждень	Розділи 1, 2
3	Розробка архітектури програми	3 тиждень	Розділи 2, 3
4	Розробка програми	4-5 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	6 тиждень	Розділ 4
6	Розробка програмної документації	7 тиждень	Розділ 5, додатки
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
8	Захист роботи	9 тиждень	

Студент(ка)

(підпис) Щербаков О.О.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Неласа Г.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 89 с., 35 рис., 4 табл., 3 дод., 15 джерел.

C#, MICROSOFT AZURE, ASP .NET CORE, ГРАФІЧНИЙ ІНТЕРФЕЙС, ЗІР, ОЧІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТРЕНУВАННЯ ОЧЕЙ.

Об'єкт дослідження – процес комп'ютерної імітації вправ для тренування очей.

Предмет дослідження – програмне забезпечення для тренування очей.

Метою роботи є розроблення програмних засобів для виконання гімнастики для очей.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C#, вебфреймворк React, хмарна платформа Microsoft Azure.

Результати. Розроблено програмне забезпечення для підтримки процесу виконання гімнастики для очей. В розробленому програмному забезпеченні реалізовано функції візуалізації комплексу вправ для тренування очей з таймером та анімацією. Розробленою програмою можуть користуватися користувачі персональних комп'ютерів з метою підтримки здоров'я очей при роботі з комп'ютером.

Висновки. Розроблено програмне забезпечення для підтримки процесу тренування очей.

Галузь використання – підтримка процесу тренування очей користувачами персональних комп'ютерів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 35 figures, 4 tables, 3 appendixes, 15 sources.

C#, MICROSOFT AZURE, ASP .NET CORE, GRAPHIC INTERFACE, SIGHT, EYES, SOFTWARE, EYE TRAINING.

The object of research is the process of computer simulation of eye training exercises.

The subject of research is software for eye training.

The aim of the work is to develop a software for performing gymnastics for the eyes.

Materials, methods and technical means: object-oriented programming, C # programming language, React web framework, Microsoft Azure cloud platform.

Results. The software of support the process of performing gymnastics for the eyes is developed. The developed software implements the functions of visualization of a complex of exercises for training of eyes with the timer and animation. The developed program can be used by personal computer users to maintain eye health when working with a computer.

Conclusions. A software for support the eye training process has been developed.

Field of usage is support the process of eye training by personal computer users.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Основні підходи щодо тренування очей.....	11
1.2 Огляд існуючих програмних продуктів для тренування очей	12
1.3 Висновки за розділом 1	16
2 Матеріали і методи.....	17
2.1 Вибір інструментарію розробки програмного забезпечення	17
2.2 Обґрунтування вибору бібліотеки React для розробки програмного забезпечення для тренування очей.....	20
2.3 Обґрунтування вибору середовища розробки програмного забезпечення для тренування очей.....	21
2.4 Розроблення структурної схеми задач програмного забезпечення для тренування очей	25
2.5 Висновки за розділом 2	26
3 Опис програми.....	28
3.1 Вимоги до програми	28
3.2 Алгоритм функціонування розробленої програми.....	29
3.3 Діаграма прецедентів програмного забезпечення для тренування очей.....	30
3.4 Основні рішення по реалізації компонентів програми	32
3.5 Висновки за розділом 3	35
4 Експлуатація та тестування програми	36
4.1 Призначення програми	36
4.2 Умови виконання програми	36
4.3 Опис роботи користувача з програмним забезпеченням для тренування очей.....	36
4.4 Результати тестування та експлуатації розробленого програмного забезпечення для тренування очей.....	37

4.6 Висновки за розділом 4	44
5 Керівництво програміста.....	45
5.1 Призначення й умови застосування програмного забезпечення для тренування очей	45
5.2 Характеристики програми.....	46
5.3 Звернення до програми.....	46
5.4 Вхідні дані.....	47
5.5 Вихідні дані.....	47
Висновки	48
Перелік джерел посилання	51
Додаток А Технічне завдання	53
Додаток Б Текст програми	58
Додаток В Слайди презентації.....	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MS	– Microsoft;
ВВПЗ	– вільне і відкрите програмне забезпечення;
ПЗ	– програмне забезпечення;
ПО	– предметна область;
ПП	– програмний продукт;
СКБД	– система керування базами даних;
ТЗ	– технічне завдання.

ВСТУП

На сьогоднішній день інформаційні технології дуже активно розвиваються, і люди все більше часу проводять перед екраном моніторів [1]-[9]. Іноді користувачам персональних комп'ютерів доводиться працювати багато годин без перерви, що негативно впливає на зорові здібності людини. Тому доцільним є щоденне виконання «гімнастики для очей» – комплексу вправ, які допомагають тримати зір у тонусі. Тож виникає потреба в створенні програмного продукту, який би дозволив виконувати такі вправи в інтерактивному режимі [1]-[9].

Актуальність створення програми пов'язана з вирішенням проблем надмірного навантаження на органи зору, а також з необхідністю більш наочного представлення вправ.

Мета роботи полягає у розробленні програмних засобів для виконання гімнастики для очей.

Програмний продукт має виконувати такі функції:

- візуалізація комплексу вправ для тренування очей з таймером та анімацією;
- надання рекомендацій щодо виконання вправ;
- автоматичний перехід між вправами, що супроводжується звуковим сигналом.

Програмний продукт матиме клієнт-серверну архітектуру, що дозволить користуватися ним одночасно декількома користувачами, а також має бути кроплатформеним і кросбраузерним.

Для досягнення мети випускної роботи бакалавра необхідно розв'язати такі завдання:

- проаналізувати предметну область щодо тренування очей за допомогою комп'ютеру;
- обрати мову програмування;
- розробити структурну схему програмного забезпечення;

- розробити функціональну схему програми;
- виконати конструювання програмного забезпечення для тренування очей;
- оформити програмну документацію на розроблену програму для тренування очей;
- протестувати розроблену програму для тренування очей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні підходи щодо тренування очей

Сьогодні багато хто проводить за комп'ютером велику частину робочого часу. Це вимагає максимальної напруги очей, від чого ми відчуваємо дискомфорт, сухість, втому. Ці, здавалося б, незначні симптоми і є перші ознаки погіршення зору [1]-[9].

Навіть якщо людина володіє стовідсотковим зором, їй все одно необхідний повноцінний відпочинок для очей. У цьому допоможе спеціальна гімнастика для очей [1]-[9].

Відомо, що очі працюють завдяки м'язам, а м'язи треба тренувати. Руху очима краще робити вранці або ввечері, перед сном. Кожну вправу варто повторювати по 5-30 разів, починати з малого, поступово збільшувати навантаження. Рухи плавні, без ривків, між вправами корисно моргнути [1]-[9].

Комплекс вправ за Коноваловою містить у собі такі вправи [1]-[3]:

- «штори» – швидке і легке кліпання очима на протязі 2 хвилин;
- «дивимося у вікно» – фокусування на предметах на екрані та за його межами;
- «великі очі» – примружування очей на 5 секунд, а потім широке їх відкривання;
- «масаж» – натискання трьома пальцями кожної руки на повіки на 1-2 секунди;
- «гідромасаж» – обполіскування очей: вранці спочатку теплою водою, потім холодною, а ввечері – навпаки;
- «картина» – примружування очей і уявлення приємної картини. Помітніший ефект досягається розтиранням долонь і прикриванням очей теплими долонями, схрестивши пальці на середині чола;
- «постріли очима» – виконання рухів очима, як зображено на рис. 1.1. для їхньої розминки.

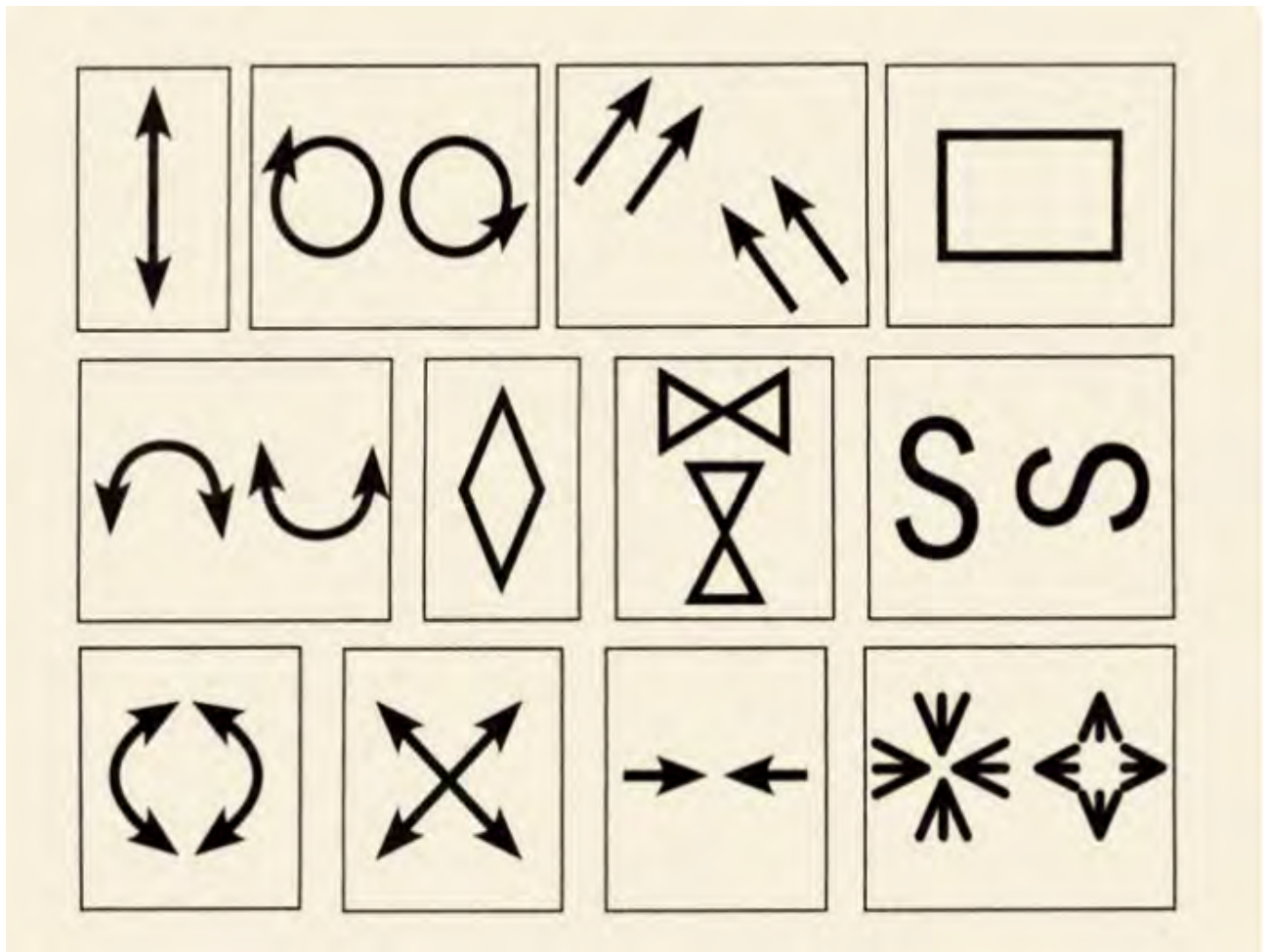


Рисунок 1.1 – Напрямок руху очей під час виконання вправи
«Постріли очима»

Виконати більшість з наведених вище вправ можливо, не відриваючись від екрану монітора (за винятком гідромасажу), тож доцільним є створення програмного продукту, який би дозволяв виконувати подібні речі [1]-[9].

1.2 Огляд існуючих програмних продуктів для тренування очей

Програмний продукт «Blimb», який представляє собою сайт blimb.su [7], дозволяє виконати комплекс вправ для тренування очей. У ньому наявний алгоритм, слідуючи якому користувач виконує цикл вправ, градієнтний фон, а переходи між вправами супроводжуються звуковими

сигналами на вибір користувача. Тривалість кожної вправи задається таймером, а для наочності рухів очей присутній рухомий курсор у вигляді круга. Крім того, сайт є мультимовним: підтримує інтерфейс користувача на українській, російській, англійській мовах. Інтерфейс сайту [7] зображений на рис 1.2.

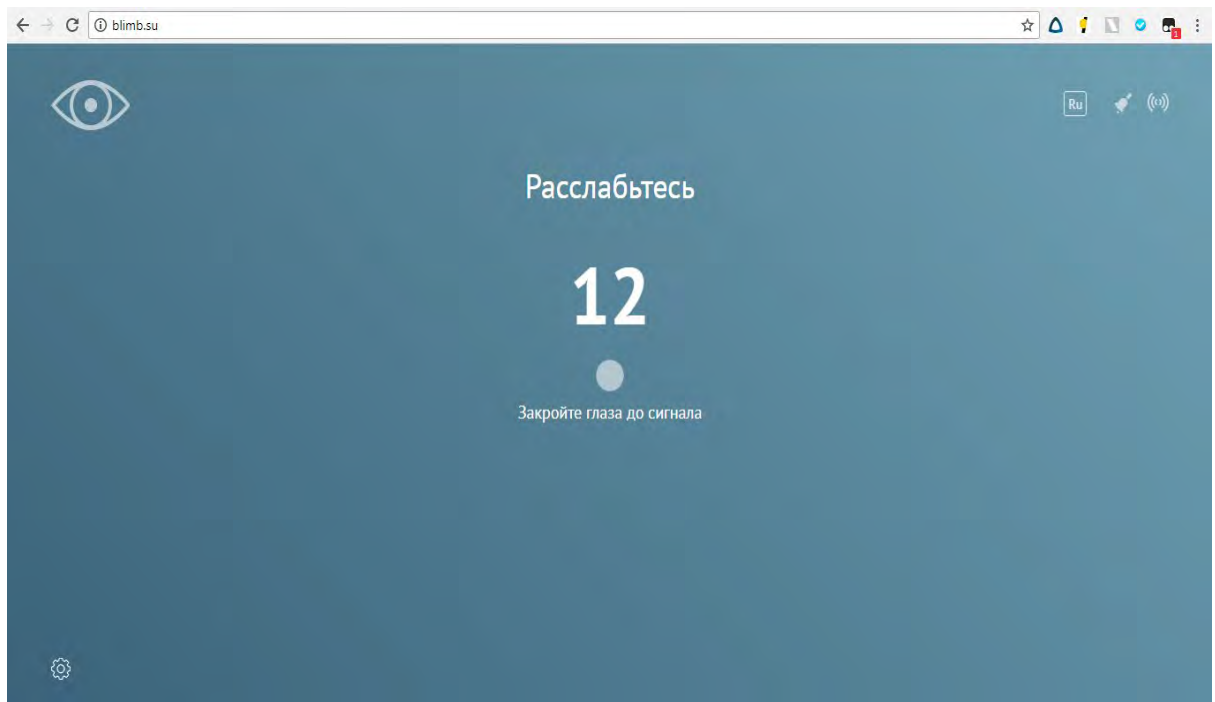


Рисунок 1.2 – Програмний продукт «Blimb»

Програмний продукт «Eyexercize», який представляє собою сайт glaz.freetonik.com [8], також дозволяє виконати гімнастику для очей, однак його функціонал підтримує всього 5 вправ, та на ньому відсутня анімація. Інтерфейс сайту [8] зображений на рис. 1.3.

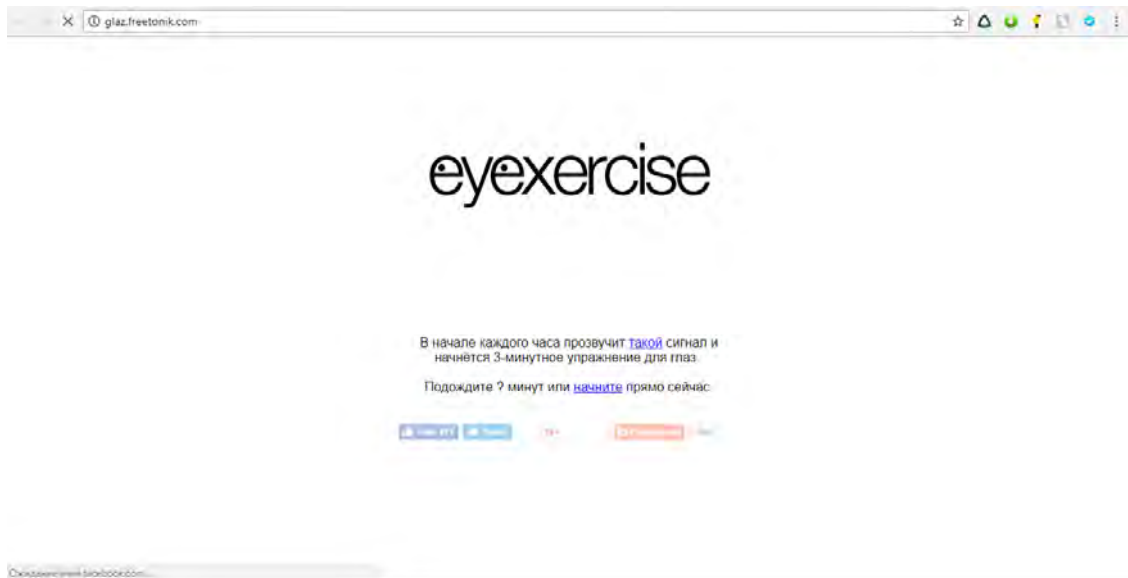


Рисунок 1.3 – Программный продукт «Eyeexercise»

Программний продукт «Упражнения для глаз» [9] представляє собою застосунок для мобільних телефонів з операційною системою Android і містить 6 найефективніших вправ, загальною тривалістю не більше 5 хвилин. Застосунок містить у собі таймер, пояснення до вправ та анімацію у вигляді великого ока. Інтерфейс застосунку [9] зображений на рис. 1.4.

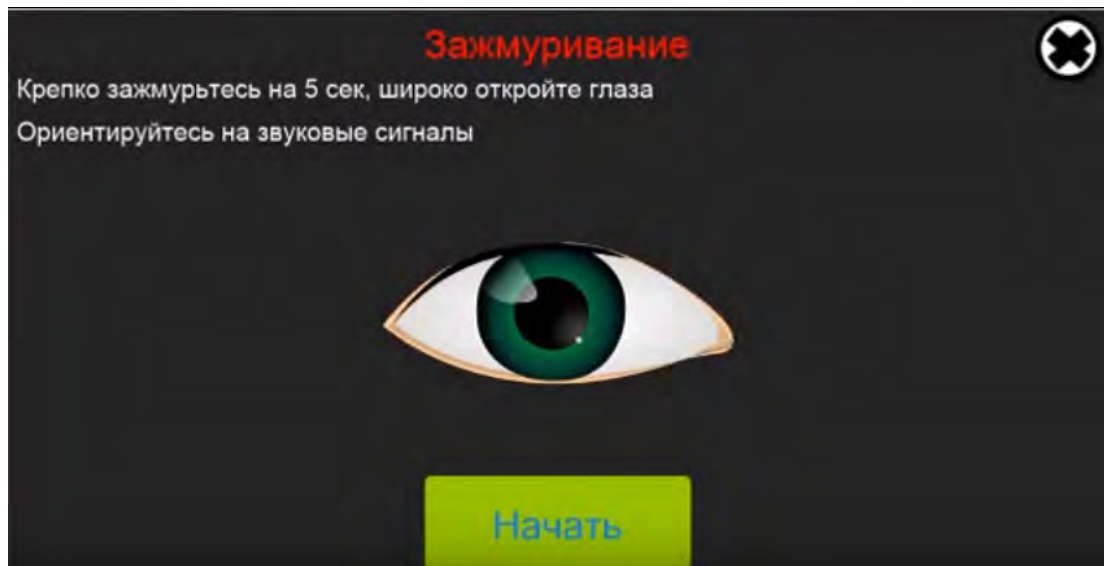


Рисунок 1.4 – Программний продукт «Упражнения для глаз»

Вище було наведено найбільш популярні на сьогоднішній день сервіси та програми, які дозволяють виконувати тренування для очей. Всі

вони, беззаперечно, мають переваги, які полягають у виконанні гімнастики для очей не відриваючись від комп'ютеру або мобільного телефону.

Але крім основних переваг, усі вони мають і недоліки. Зокрема, сервіси «Blimb» та «Eyexercise» потребують постійного підключення до мережі Інтернет, на відміну від ПП «Упражнения для глаз». Сервіс «Eyexercise» не має анімаційних елементів, ПП «Упражнения для глаз» не містить таймеру, а сервіс «Blimb» хоч і має більший функціонал, однак все ж не повний, у порівнянні зі списком вправ, зазначених у підрозділі 1.1.

Підсумовуючи вищезазначене, у табл.1.1 наведено порівняння важливих характеристик вже існуючих програмних продуктів та програми, яка буде розроблена в ході виконання роботи.

Таблиця 1.1 – Порівняння програмних продуктів

Можливості	«Blimb»	«Eyexercise»	«Упражнения для глаз»	Розроблюваний ПП
Виконання тренування	+	+	+	+
Таймер	+	+	—	+
Рекомендації	+	+	+	+
Звуковий супровід	+	+	+	+
Анімаційні елементи	+	—	+	+
Робота через Інтернет	+	+	—	+
Клієнт-серверна архітектура	+	+	+	+
Кросплатформеність	+	+	—	+
Повний комплекс вправ	—	—	—	+
Додаткові елементи інтерфейсу	+	+	—	—

Зважаючи на наведену порівняльну характеристику у табл.1.1 та виявлені недоліки аналогів, доцільно розробити власний програмний продукт, який буде забезпечувати необхідну функціональність згідно з вимогами предметної області.

ПП повинен забезпечувати виконання нижче перерахованих функцій:

- можливість користувачу виконувати гімнастику для очей;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до ПП багатьох користувачів з пристроїв на різних операційних системах.

1.3 Висновки за розділом 1

Проаналізовано проблему створення програмних продуктів для тренування очей. Проаналізовано програмне забезпечення для автоматизації процесу рекомендацій щодо виконання вправ для тренування очей. Виявлено, що найбільш популярні на сьогоднішній день сервіси та програми, які дозволяють виконувати тренування для очей, беззаперечно, мають переваги, які полягають у виконанні гімнастики для очей, не відриваючись від комп'ютеру або мобільного телефону. Проте, сервіси «Blimb» та «Eyexercise» потребують постійного підключення до мережі Інтернет, на відміну від програмного продукту «Упражнения для глаз». Сервіс «Eyexercise» не має анімаційних елементів, програмний продукт «Упражнения для глаз» не містить таймеру, а сервіс «Blimb», хоч і має більший функціонал за інші розглянуті аналоги, однак все ж не повний, у порівнянні зі списком вправ, зазначених у підрозділі 1.1.

Отже, за результатом аналізу програм-аналогів було прийнято рішення про доцільність розробки власного програмного забезпечення для тренування очей, а також сформовано основні функціональні вимоги до розроблюваної програми.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір інструментарію розробки програмного забезпечення

У якості мов програмування було обрано C# [10] та JavaScript [11], функції бібліотеки React [12], а також фреймворк-середовище розробки Microsoft Visual Studio [13]-[14] і хмарний сервіс Azure [15].

C# – це об'єктно-орієнтована мова програмування, яка характеризується принципово важливими рішеннями, що реалізовані корпорацією Microsoft, до яких можна віднести наступні [10]:

- компонентно-орієнтований підхід до програмування;
- властивості як засіб інкапсуляції даних;
- обробка подій;
- уніфікована система типізації;
- індексатори;
- механізми boxing і unboxing для перетворення типів;
- атрибути (засіб оперування метаданими в СОМ-моделі).

До переваг C # можна віднести [10]:

- C # є повністю об'єктно-орієнтованою мовою, що має риси C/C++;
- єдиний каркас розробки – Framework .Net;
- підтримка FCL та CLR;
- підтримка спадкування та універсалізації.

Зважаючи на вище зазначений опис мов програмування для реалізації Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато що від своїх попередників – мов C++, Delphi, Модула і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при

розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++) [10].

C# є дуже близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу роззосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечні речі (типу вказівників без контролю виходу за межі). Для роззосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектуального пристрою [10].

Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком) [10].

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET [10].

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки ПЗ та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються MS Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки ПЗ та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з

графічним інтерфейсом, в тому числі з підтримкою технологій WPF, та нової – UWP, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight [10].

Порівняльна характеристика мов програмування наведена в таблиці 2.1.

Таблиця 2.1 – Порівняння мов програмування

Критерій порівняння	PHP	C#	Java
Засоби для створення графічного інтерфейсу	±	+	±
Ефективні засоби об'єктно-орієнтованого програмування	±	+	+
Можливість роботи з прототипами	–	+	–
Підтримка парадигми процедурного програмування	+	+	–
Зручність використання для створення програмного забезпечення для тренування очей	–	+	±
Можливість роботи з базами даних	+	+	+

Таким чином, для розробки програмного забезпечення для тренування очей доцільно обрати мову C#, оскільки ця мова дозволяє ефективно створювати графічний інтерфейс користувача, є об'єктно-орієнтованою мовою програмування, а також характеризується зручністю використання для створення програмного забезпечення для тренування очей.

Також для розробки програмного забезпечення для тренування очей використовується мова JavaScript (JS), яка є динамічною, об'єктно-орієнтованою прототипною мовою програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з

користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [11].

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу [11].

Мова JavaScript використовується для [11]:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

При використанні в рамках технології DHTML JavaScript код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги `<script></script>` з обов'язковим по специфікації HTML 4.01 атрибутом `type="text/javascript"`, хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript [11].

2.2 Обґрунтування вибору бібліотеки React для розробки програмного забезпечення для тренування очей

Для розробки програмного забезпечення для тренування очей обрано бібліотеку React мови JavaScript. React (React.js, ReactJS) – відкрита

JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників [12].

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає представленню у шаблоні модель-представлення-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків. Як бібліотека інтерфейсу користувача React часто використовується разом з іншими бібліотеками, такими як Redux [12].

Компоненти React зазвичай написані на JSX. Код написаний на JSX компілюється у виклики методів бібліотеки React. Розробники можуть так само писати на чистому JavaScript. JSX нагадує іншу мову, яку створили у компанії Фейсбук для розширення PHP, XHP [12].

Код, написаний у JSX, потребує перетворення за допомогою такого інструменту, як Babel, для того, щоб його могли зрозуміти веб-браузери. Ця обробка, як правило, виконується під час процесу збірки, перш ніж програма буде запущена [12].

2.3 Обґрунтування вибору середовища розробки програмного забезпечення для тренування очей

Середовище розробки Visual Studio було обрано для створення програмного забезпечення для тренування очей як перевіреним часом програмний продукт [13], [14].

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки ПЗ та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби [13], [14].

Головні переваги даного середовища [13], [14]:

- відкритість для мов програмування;
- принципово новий підхід до побудови каркаса середовища - Framework .Net;
- включає редактор вихідного коду з підтримкою технології IntelliSense і можливістю рефакторингу коду;
- редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних;
- підтримка систем контролю версій вихідного коду.

Visual Studio включає .NET Framework. Версія мала незначні зміни у інтерфейсі інсталятора та самого засобу розробки. Основну увагу було зосереджено на підтримці та розвитку нового кросплатформеного фреймворку .Net Core та заохочення створення додатків на його основі. В більшій мірі це стосується створення бізнес логіки веб-додатків на основі ASP.NET Core [13], [14].

ASP.NET Core – це крос-платформена, високопродуктивна платформа з відкритим кодом для створення сучасних, хмарних, підключених до мережі Інтернет додатків. За допомогою ASP.NET Core можливо [13], [14]:

- створювати веб-програми та служби, додатки IoT і мобільні додатки.
- розроблювати програмні продукти для Windows, MacOS та Linux.
- розгорнути веб-додатки на хмарному або локальному сервері.
- запускати додатки на .NET Core або .NET Framework.

Порівняльна характеристика найпоширеніших середовищ розробки для мови C# наведена у таблиці 2.2.

Таблиця 2.2 – Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Microsoft VS	SharpDevelop	MonoDevelop
Зручність для розробки програмного забезпечення для тренування очей	+	+	—
Кросплатформність	+	—	+
Підтримка репозиторіїв	+	+	—
Швидкість модернізації розроблюваного програмного забезпечення	+	—	+
Наявність візуального конструктора	+	+	+

Зважаючи на наведену порівняльну характеристику у табл.2.1 та виявлені недоліки таких систем, як SharpDevelop та MonoDevelop, було вирішено обрати саме інтегроване середовище розробки Visual Studio.

Також для розробки програмного забезпечення для тренування очей буде використовуватися платформа Microsoft Azure – це хмарна платформа та інфраструктура корпорації Microsoft, призначена для розробників застосунків хмарних обчислень і покликана спростити процес створення он-лайн додатків [15].

Microsoft Azure дозволяє створювати додатки як за допомогою Microsoft .NET Framework і Visual Studio, так і за допомогою інших інструментів. Операційна система працює на серверах Microsoft, доступ до неї можна отримати за протоколами HTTP, Representational State Transfer (REST), WS-* і Atom Publishing Protocol (AtomPub) [15].

Концепція хмарних обчислень – це використання обчислювальних потужностей, дискового простору і каналів зв'язку «обчислювальної хмари» для виконання трудомістких завдань. Навантаження між комп'ютерами, що входять в цю хмару, розподіляється автоматично. Більшість хмарних застосунків працюють у браузері [15].

За допомогою MS Azure процес публікації додатків стає аналогічним запуску в режимі зневадження. Тобто, попередньо налаштувавши MS Azure і отримавши підписку (достатньо безкоштовної пробної версії), розробку лише необхідно вказати адресу сайту і натиснути кнопку опублікувати [15]. Інтерфейс Microsoft Azure наведений на рис. 2.1.

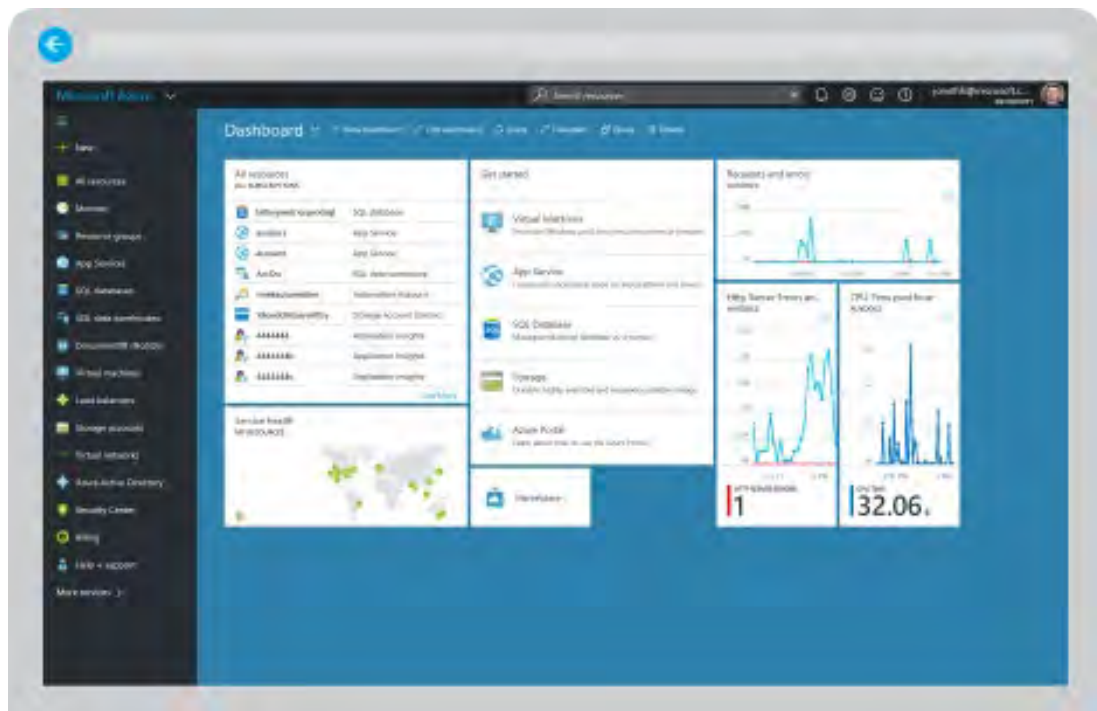


Рисунок 2.1 – Інтерфейс Microsoft Azure

2.4 Розроблення структурної схеми задач програмного забезпечення для тренування очей

На рис 2.2 представлена структурна схема задач програмного продукту для тренування очей з точки зору виконуваних функціональних задач.

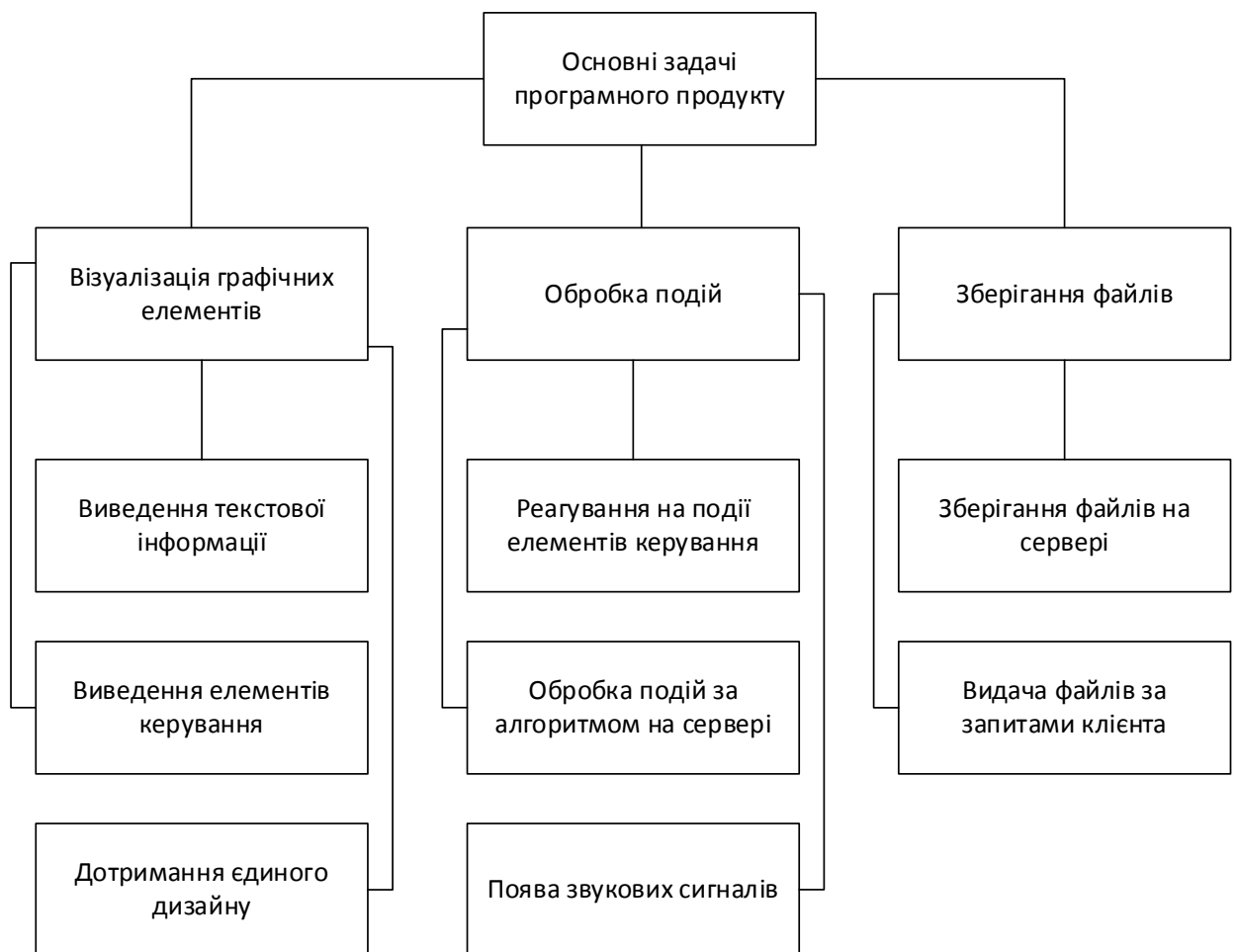


Рисунок 2.2 – Структурна схема задач програмного продукту

Процес функціонування програмного продукту складається з таких фаз:

- фаза ініціалізації. Користувач вводить адресу сайту у рядку пошуку свого браузера і на сервер йде запит на пошук файлу «index.html»;
- фаза роботи користувача з програмою. Користувач може здійснювати запити до серверу, зокрема, щоб почати тренування, переглянути сторінку з рекомендаціями, або оновити сторінку;
- фаза виконання алгоритму. Якщо користувач не оновлює сторінку, то після початку тренування на екрані з'являється динамічний вміст сторінки – в залежності від вправи змінюється покази таймеру, рекомендації та положення анімаційного елементу.

2.5 Висновки за розділом 2

Для розробки програмного забезпечення для тренування очей обрано мову C#, оскільки ця мова дозволяє ефективно створювати графічний інтерфейс користувача, є об'єктно-орієнтованою мовою програмування, а також характеризується зручністю використання для створення програмного забезпечення для тренування очей. Також для розробки програмного забезпечення для тренування очей використовується мова JavaScript (JS), яка є динамічною, об'єктно-орієнтованою прототипною мовою програмування.

Для розробки програмного забезпечення для тренування очей обрано бібліотеку React мови JavaScript, яка є відкритою JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків.

Проаналізовано середовища розробки програмного забезпечення. Зважаючи на наведену порівняльну характеристику у табл.2.1 та виявлені недоліки таких систем, як SharpDevelop та MonoDevelop, було вирішено обрати саме інтегроване середовище розробки Visual Studio.

Також для розробки програмного забезпечення для тренування очей буде використовуватися платформа Microsoft Azure – це хмарна платформа

та інфраструктура корпорації Microsoft, призначена для розробників застосунків хмарних обчислень і покликана спростити процес створення он-лайн додатків.

Визначено загальну структуру задач програмного продукту, що дозволить розробити програмне забезпечення відповідно до проаналізованих та виявлених вимог.

3 ОПИС ПРОГРАМИ

3.1 Вимоги до програми

Програмний продукт повинен надавати можливість одночасного доступу багатьох користувачів, рекомендацій щодо виконання вправ, автоматичного переходу між вправами, що супроводжується звуковим сигналом.

Програмний продукт для тренування очей має виконувати такі функції:

- користувач може за його допомогою виконувати гімнастику для очей;
- наявність таких вправ, як: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- наявність таймеру та анімації для кожної вправи;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до ПП багатьох користувачів з пристроїв на різних операційних системах.

До вхідних даних програми відносяться запити користувача: натискання кнопки старту або переходу на сторінку із загальними рекомендаціями.

Вихідні дані програми організовані у вигляді інтерфейсу програми, що включає рекомендації до кожної з вправ, таймер та анімаційні елементи.

Інтерфейс користувача повинен забезпечити просту й ефективну взаємодію користувача з програмним продуктом.

Таким чином, розроблюване програмне забезпечення повинно дозволяти виконувати гімнастику для очей в інтерактивному режимі з наочним представлення вправ.

3.2 Алгоритм функціонування розробленої програми

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів. Клієнт-серверна система представлена на рис. 3.1.

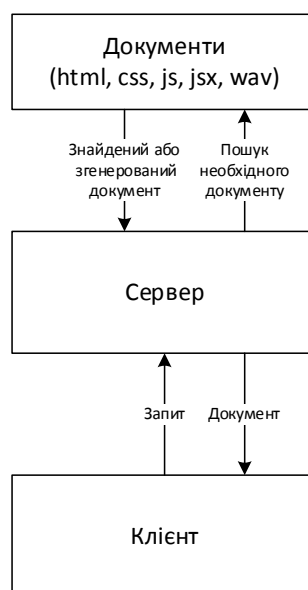


Рисунок 3.1 – Функціональна схема програми

Алгоритм функціонування програми повністю відповідає розглянутому вище прецеденту «Тренування». Функціональна схема програми на рис.3.2.

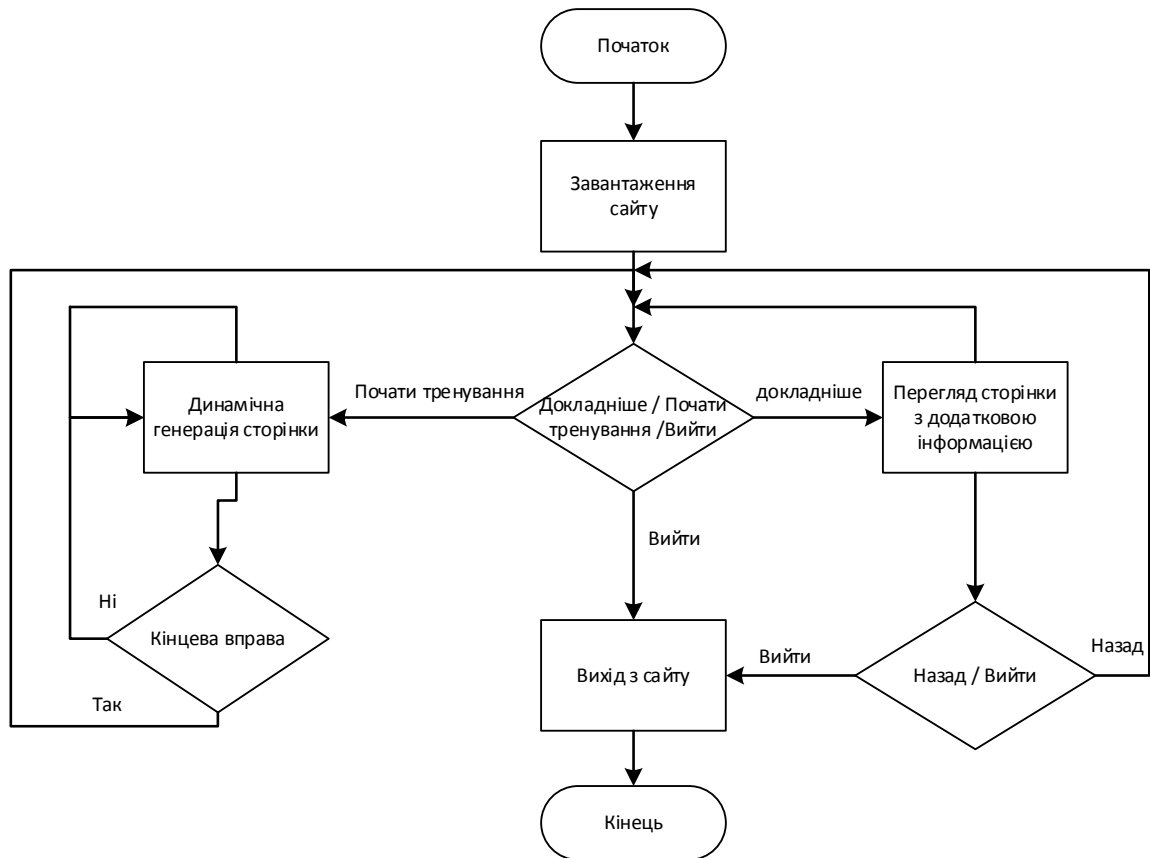


Рисунок 3.2 – Функціональна схема для клієнта

3.3 Діаграма прецедентів програмного забезпечення для тренування очей

У розробленій системі «EyeTrainer» основна мета користувача – здійснити гімнастику для очей. Тому розглянемо прецедент «Тренування».

Прецедент «Тренування».

Основні виконавці – користувач, система «EyeTrainer», сервер.

Зацікавлена особа – користувач.

Передумова – наявність введеної у рядок пошуку адреси сайту.

Постумова – закінчення тренування.

Основний успішний сценарій для даного прецеденту такий:

- користувач після введення адреси натискає кнопку enter;
- сервер у відповідь на запит повертає сторінку index.html;
- користувач ознайомлюється із загальними рекомендаціями, натиснувши кнопку «Докладніше». Для системи це означає запит до серверу, який повертає сторінку about.html;
- користувач розпочинає тренування;
- користувач виконує вправи: для системи це означає запит до серверу, який динамічно відображує сторінку index.html за алгоритмом з файлу index.jsx. Динамічний вміст сторінки полягає у зміні значення секунд таймеру та анімації для кожної вправи, а саме для вправ: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- закінчивши тренування, користувач покидає сайт.

Для розробленої системи «EyeTrainer» діаграма виглядає так, як показано на рис. 3.3.

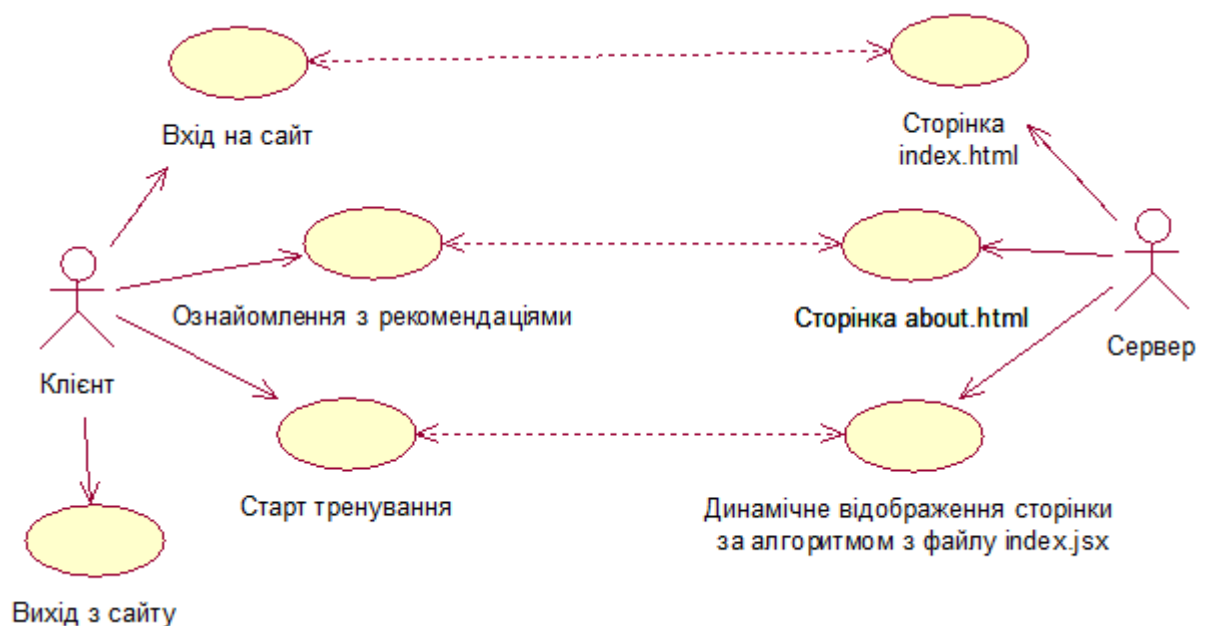


Рисунок 3.3 – Діаграма прецеденту «Тренування»

3.4 Основні рішення по реалізації компонентів програми

Розроблений програмний продукт складається з C#, JSX, CSS та HTML файлів, а також файлів налаштування проекту, які створюються автоматично під час вибору шаблону ASP .NET Core -> MVC -> React.

Файли проекту призначені для виконання таких функцій:

- стартова сторінка `index.cshtml`: призначена для візуалізації стартової сторінки сайту і містить посилання на React-компонент;
- налаштування стилів сайту `Styles.css`: містить всі стилі сайту у вигляді CSS-класів, включаючи опис анімаційних елементів сайту у вигляді `keyframes`-методів;
- налаштування динамічного наповнення сайту у файлі `index.jsx`: містить компонент React з методами обробки загального алгоритму появи вправ, таймеру, анімації, звуків. Компонент представляє собою клас з методами обробки значення таймеру, а також метод `render()`, який повертає HTML-розмітку сторінки в залежності від поточного значення таймера.

Файли проекту та їх короткий опис наведені у табл.3.1.

Таблиця 3.1 – Файли проекту програмного продукту

Ім'я файлу	Опис
<code>Program.cs</code>	Створення веб-хосту і його запуск
<code>Startup.cs</code>	Налаштування конфігурацій та сервісів
<code>index.cshtml</code>	Стартова сторінка
<code>Styles.css</code>	Налаштування стилів сайту
<code>index.jsx</code>	Налаштування динамічного наповнення сайту: загального алгоритму появи вправ, таймеру, анімації, звуків

Створення веб-хосту, налаштування конфігурацій. За створення веб-хосту відповідає клас `Program` з файлу `Program.cs`. Він містить статичний метод `BuildWebHost`, який налаштовує, створює, а потім повертає посилання на `Host`. Цей метод містить виклик методу `CreateDefaultBuilder` класу

помічника WebHost, який відповідає за створення екземпляра WebhostBuilder з потрібною конфігурацією.

CreateDefaultBuilder повідомляє хосту використовувати веб-сервер Kestrel. Kestrel – це крос-платформений керований HTTP-сервер. Цей сервер дозволяє додатку ASP.NET Core працювати на ОС, відмінній від Windows[10]. Крім того, CreateDefaultBuilder відповідає також за встановлення поточного каталогу як кореневого каталогу програми, підтримки логування, налаштування інтеграції Kestrel з сервером IIS і повертання об'єкту збірки.

BuildWebHost викликає також метод UseStartup, якому передається посилання на клас Startup з файлу Startup.cs. Клас Startup містить метод ConfigureServices, який налаштовує служби для ін'єкцій залежності та метод Configure, який налаштовує обробку всіх HTTP-запитів, поданих до програми.

Метод побудови WebHostBuilder створює об'єкт WebHost, на якому розміщується прикладна програма, а метод Run з методу Main запускає створений веб-хост і починає слухати HTTP-запити.

Вміст стартової сторінки. Конфігурації, які задаються у класі HomeController з файлу HomeController.cs, повертають відповідні представлення з cshtml-файлів каталогу Views. Зокрема метод Index() класу HomeController викликає вміст файлу index.cshtml. Вміст сторінки index.html, яка буде сформована як результат запуску файлу index.cshtml, містить у собі посилання на файл зі стилями, а також на React-компонент.

Динамічне наповнення сайту. Оскільки сайт представляє собою імітацію рухів під час гімнастики очей, то необхідно динамічно змінювати вміст сторінки. Для цього у розробленій програмі було створено клас EyeTop у файлі index.jsx, який успадковує дані та методи класу React.Component. Основні методи класу EyeTop такі:

- `constructor(props)` – представляє собою конструктор класу, в якому ініціалізуються змінні з кількістю секунд, протягом яких виконується та чи інша вправа;
- `tick()` – метод, що відповідає за логіку перемикавання між вправами в залежності від значень таймера;
- `startTimerEye()` / `startTimerStart()` / `startTimerBlocks()` – встановлення інтервалу перемикавання в залежності від типу елемента на 50 / 70 / 1000 мілісекунд відповідно;
- `playAudio()` – метод для виклику звукового сигналу у момент переключення вправ;
- `render()` – головний метод, що повертає поточний стан тренування – html-розмітку для поточної вправи, який включає в себе відповідне значення таймера і положення анімаційного елемента у відповідний момент часу.

Асоціювати розроблений клас з елементом у html можна, надавши елементу певний ідентифікатор, наприклад `<div id="eyetop"></div>`, а у файлі, де знаходиться розроблений клас, зазначити `ReactDOM.render(<EyeTop />, document.getElementById("eyetop"))`.

Стили сайту. Усі стилі сайту містяться у файлі `Styles.css` у вигляді `css`-класів. Стили відіграють важливу роль у дизайні сторінки, виконуючи такі функції:

- плавне «закривання ока» на головній сторінці – схематичний рисунок ока, що представляє собою набір зображень, які змінюються з періодом у 50 мілісекунд, що створює ефект динамічності. Реалізовано шляхом використання методу `render()` з класу `EyeTop` і набору класів з різними позиціями фону;
- реалізація стилів тексту, таймеру, кнопок, посилань;
- градієнтизація фону за допомогою правил `@keyframes` і властивості `animation`;

- створення класів, які засобами CSS створюють динамічний елемент
- коло, та заданих для кожної вправи певних властивостей animation і відповідних @keyframes правил.

3.5 Висновки за розділом 3

Проаналізовано основні вимоги до ПП для тренування очей.

Створено програмний продукт, який повністю реалізує поставлену задачу. Тобто, розроблений ПП реалізує основний алгоритм гімнастики для очей.

Крім того, було обрано механізм реалізації цих алгоритмів за допомогою діаграми прецедентів та функціональної схеми. А також були обрані елементи керування (підказки, звуки, анімація), з якими користувачу буде легше взаємодіяти з програмним продуктом.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Програмний продукт для тренування очей призначений для зняття нервового напруження на органи зору шляхом виконання гімнастики для очей і представляє собою інтерфейс для виконання комплексу вправ.

Програмний продукт надає можливість одночасного доступу багатьох користувачів, рекомендацій щодо виконання вправ, автоматичного переходу між вправами, що супроводжується звуковим сигналом.

4.2 Умови виконання програми

Для відсутності проблем функціонування системи на комп'ютері користувача повинні виконуватися такі умови до апаратної та програмної частини:

- процесор із тактовою частотою 2 GHz та більше;
- наявність IBM-сумісного персонального комп'ютеру;
- оперативна пам'ять 1 ГБ та більше;
- наявність будь-якої з операційних систем Windows, Linux, Mac;
- наявність браузеру для виходу в Інтернет (бажано, Google Chrome);
- вільний простір на жорсткому диску не менше, ніж 15 Мб.

4.3 Опис роботи користувача з програмним забезпеченням для тренування очей

Для запуску програму необхідно підключитись до мережі Інтернет, ввести у рядок браузеру відповідну адресу і, в залежності від швидкості з'єднання почекати декілька секунд. За потреби, натиснути кнопку «Докладніше» для ознайомлення із загальними рекомендаціями.

Для початку виконання гімнастики для очей необхідно натиснути кнопку «Розпочати тренування», далі – «Почати», і впродовж трьох хвилин двадцяти п'яти секунд виконувати зазначені вправи. Після закінчення вправ їх можна повторити, знов натиснувши кнопку «Розпочати тренування», або закрити вкладку браузера.

4.4 Результати тестування та експлуатації розробленого програмного забезпечення для тренування очей

Тестування програмного забезпечення для тренування очей буде проведене за таким сценарієм:

- користувач запускає браузер і вводить у відповідний рядок адреси програми, натискає кнопку enter;
- користувач ознайомлюється із загальними рекомендаціями, натиснувши кнопку «Докладніше»;
- користувач розпочинає використання програмного забезпечення для тренування очей;
- користувач виконує вправи: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- закінчивши тренування очей, користувач покидає сайт.

Якщо в налаштуваннях Azure допустити помилки, то сайт буде недоступним. Результат помилки зображений на рис 4.1

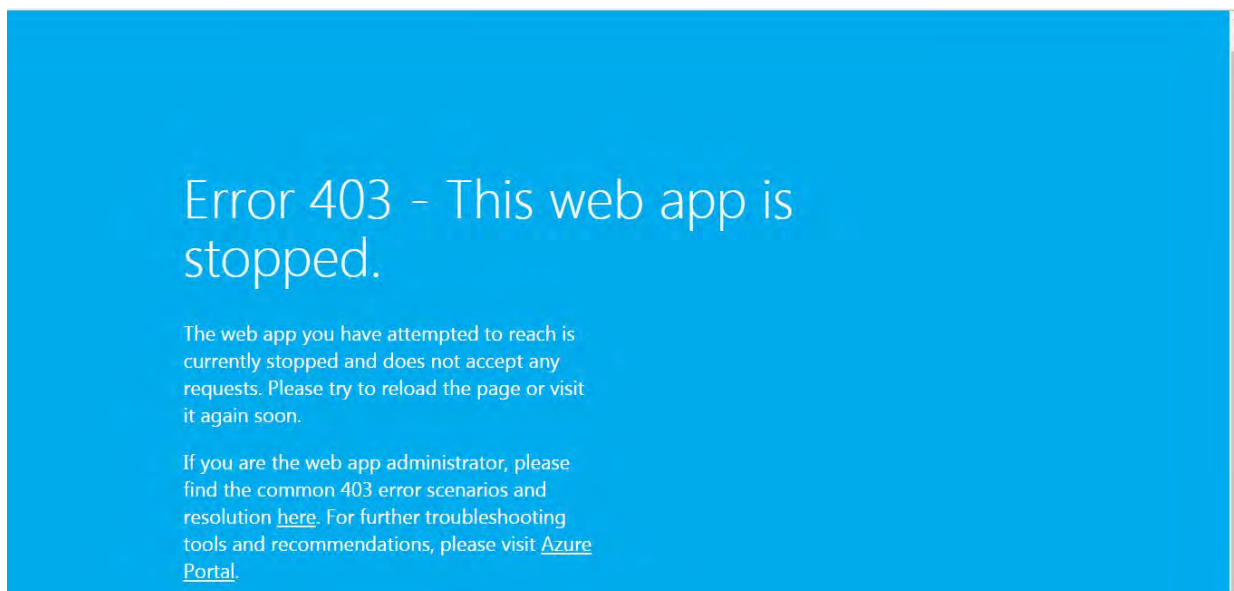


Рисунок 4.1 – Помилка при входженні на сайт

Стартова сторінка сайту зображена на рис 4.2.

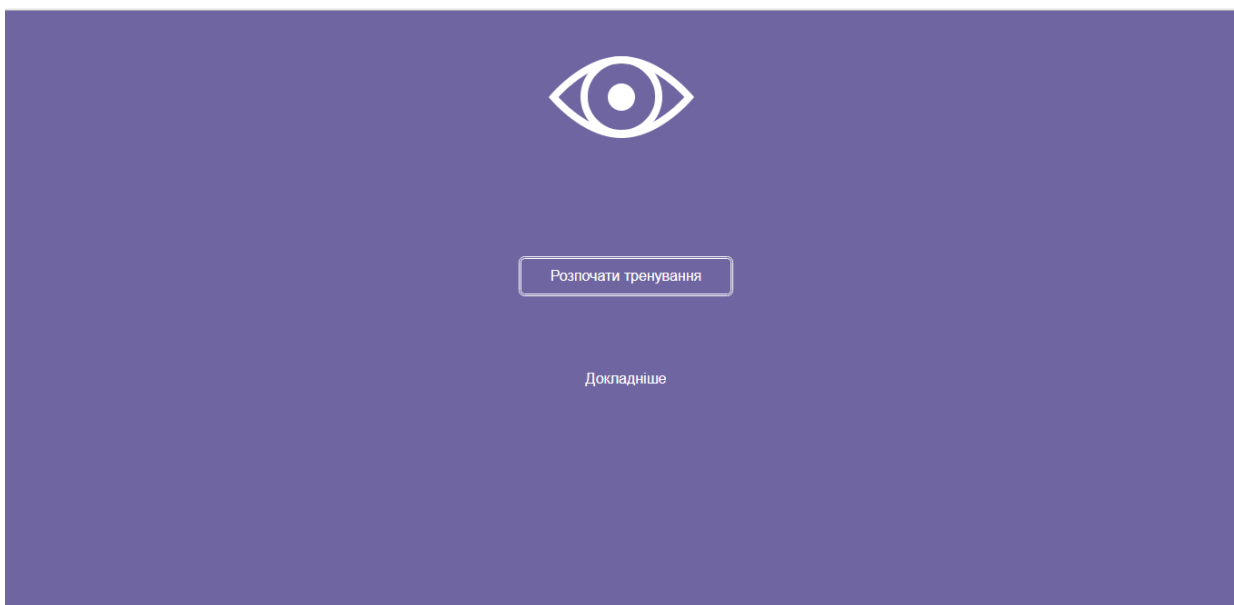


Рисунок 4.2 – Стартова сторінка сайту

Після натискання кнопки «Докладніше» відбувається перехід на сторінку `about.html` (рис 4.3).

Вправи для очей під час роботи за монітором потрібні не менше, ніж розминка спини після важкого робочого дня. 2-3 хвилини вправ допомагають повернути тонус м'язам очей. Найбільшого ефекту Ви досягнете, якщо після вправи зробите масаж шиї та відпочиньте декілька хвилин.

Рекомендації

Перед вправою потрібно розслабити очі. Для цього прикрийте їх руками на пів хвилини, або зробіть пальмінг.

Закрийте очі, та прикрийте їх руками, щоб не проходило світло. Розслабтесь.

Намагайтеся виконувати вправи кожні пів години.

В комплекті, набір вправ, під час яких потрібно дивитися у напрямках, вказаних курсором.

Білий кружечок вказує тільки напрямок - погляд зміщуйте до упору, але не занадто.

У крайніх точках затримуйте погляд на секунду.

Рисунок 4.3 – Сторінка із загальними рекомендаціями

Далі на головній сторінці зображені короткі рекомендації (рис 4.4).

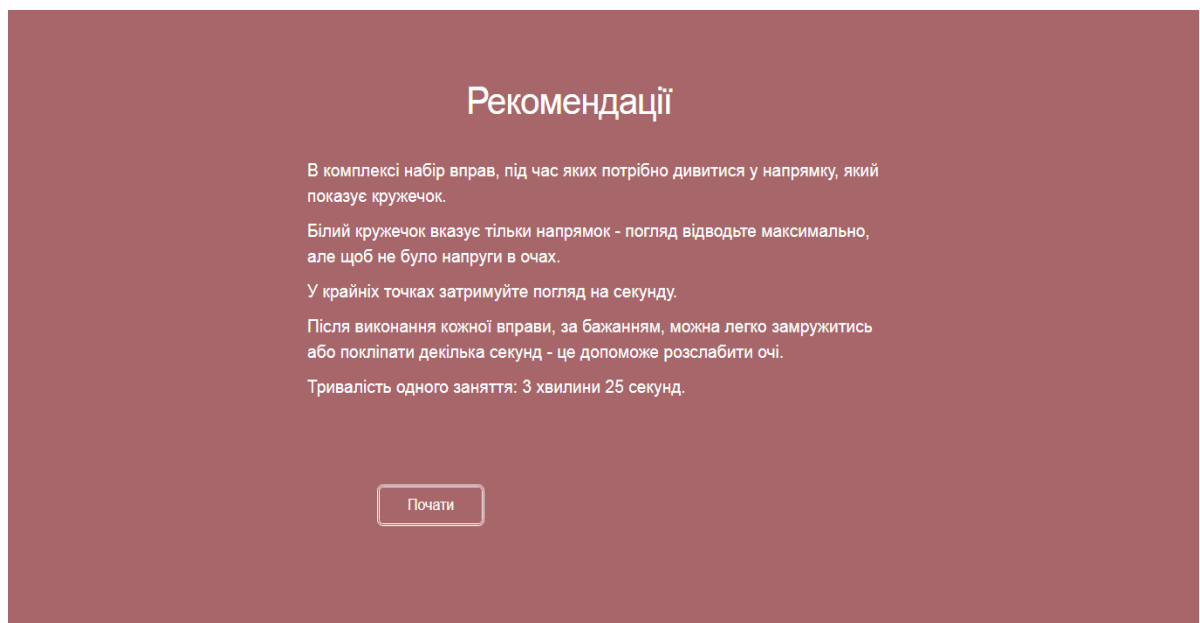


Рисунок 4.4 – Сторінка із короткими рекомендаціями перед стартом

Далі на рис 4.5 наведена перша вправа.



Рисунок 4.5 – Сторінка із першою вправою

Далі зображені вправи «вгору-вниз» (рис 4.6), «вліво-вправо» (рис 4.7), «по діагоналі» (рис. 4.8), «за годинниковою стрілкою» (рис. 4.9), «по спіралі» (рис. 4.10), «далеко-близько» (рис. 4.11).

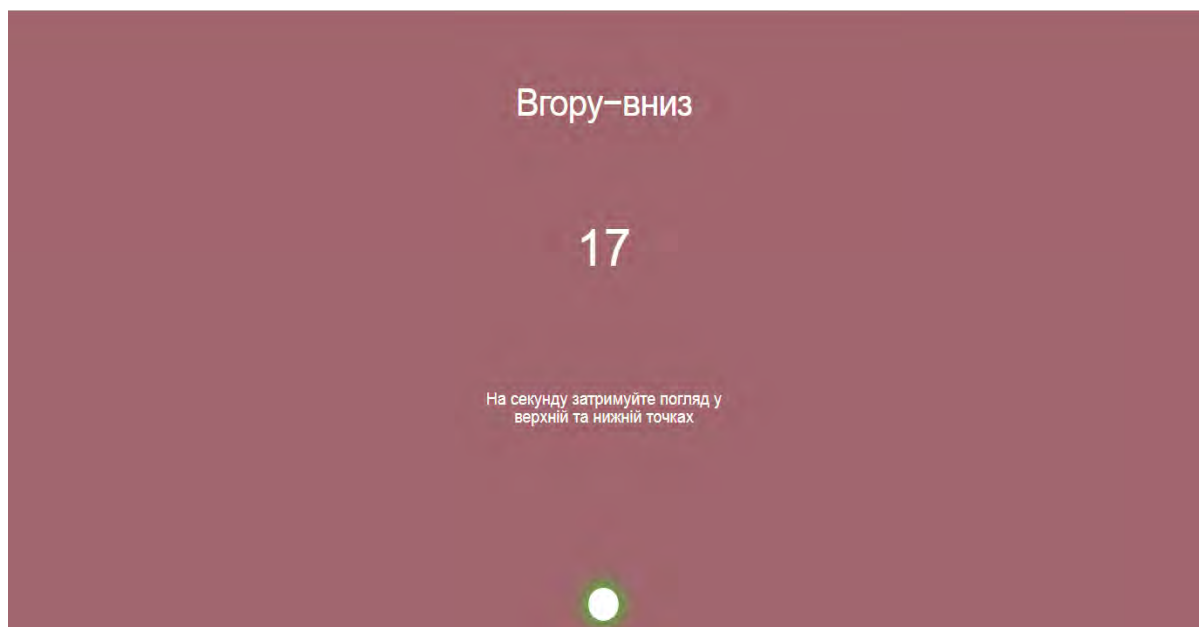


Рисунок 4.6 – Сторінка із вправою «вгору-вниз»

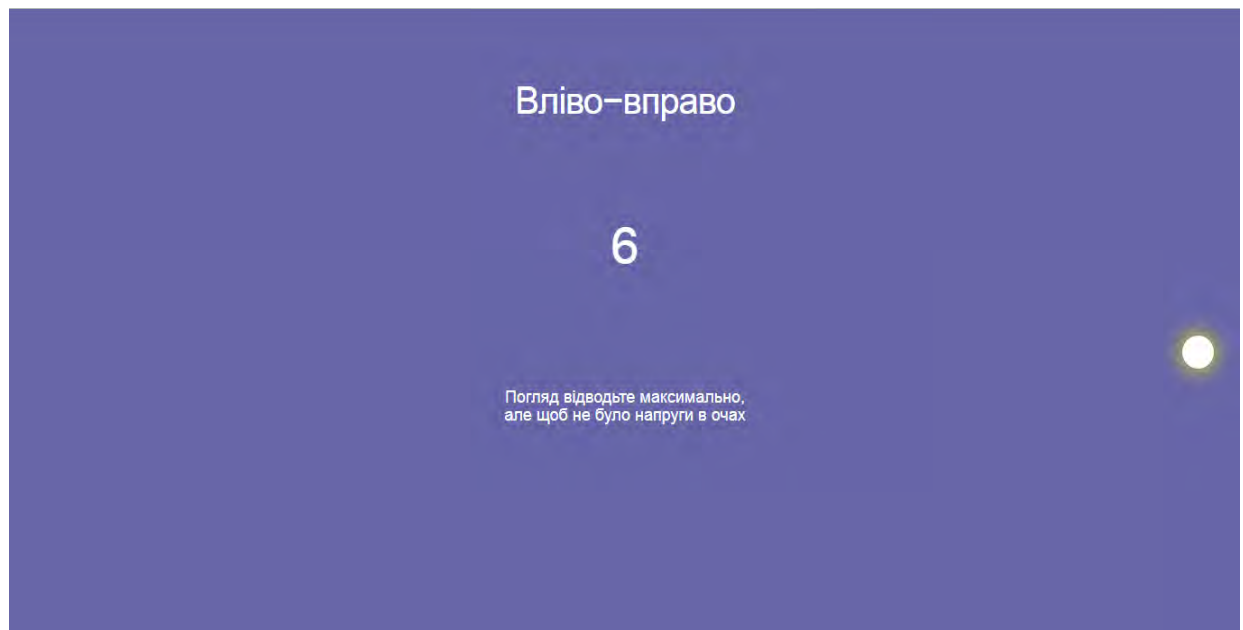


Рисунок 4.7 – Сторінка із вправою «вліво-вправо»

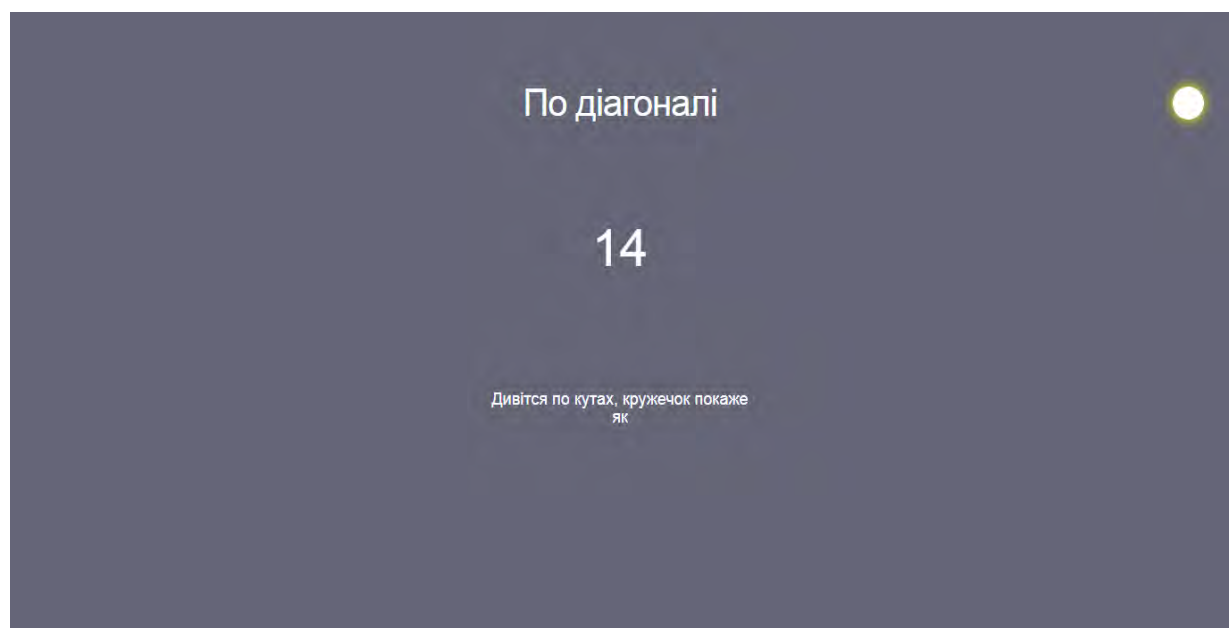


Рисунок 4.8 – Сторінка із вправою «по діагоналі»

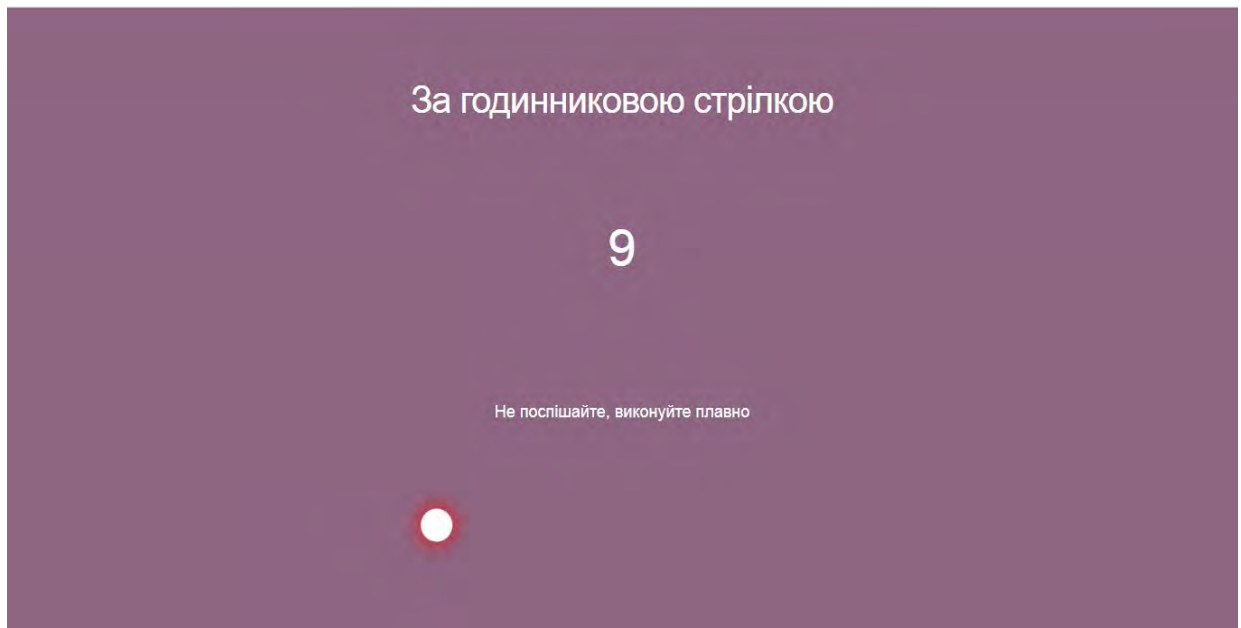


Рисунок 4.9 – Сторінка із вправою «за годинниковою стрілкою»

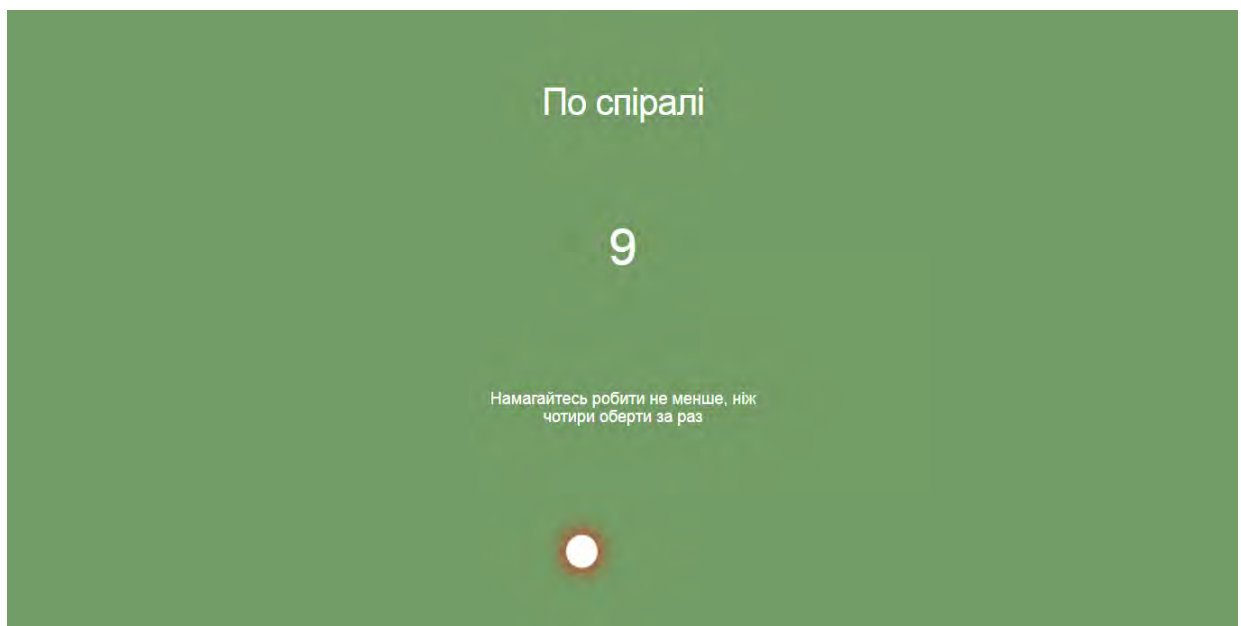


Рисунок 4.10 – Сторінка із вправою «по спіралі»



Рисунок 4.11 – Сторінка із вправою «далеко-близько»

По закінченню виконання вправ з'являється вітання із завершенням тренування (рис. 4.12).

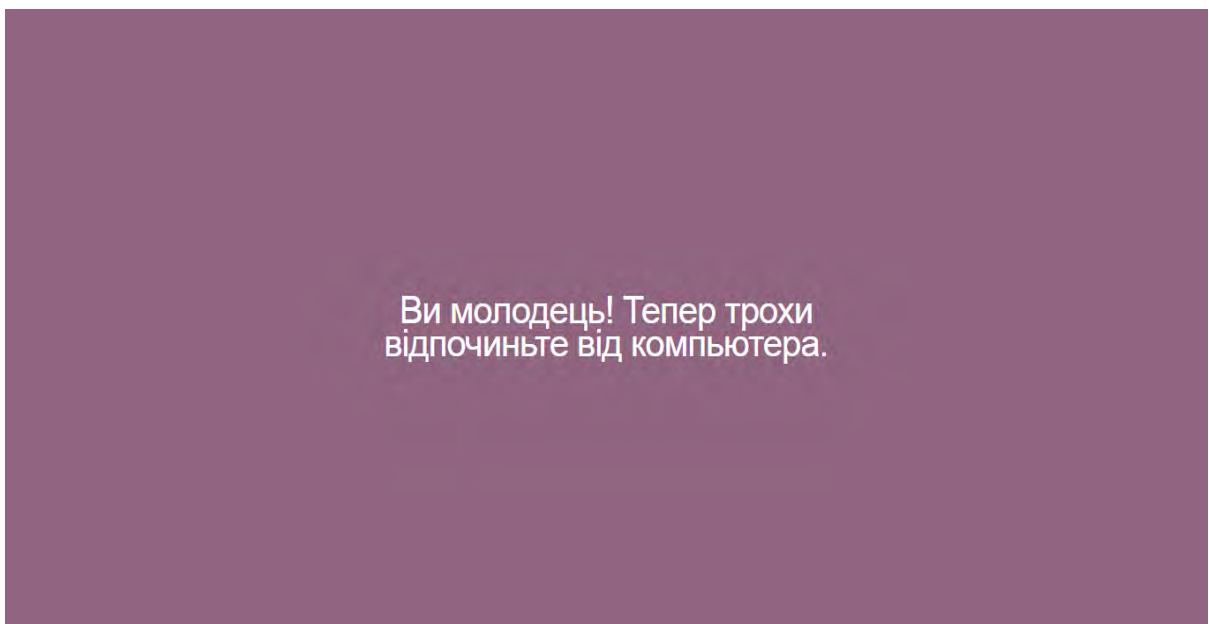


Рисунок 4.12 – Сторінка із завершенням тренування

Отже, результати тестування та використання програмного забезпечення для тренування очей підтвердили його працездатність.

4.6 Висновки за розділом 4

Описано основні умови виконання програмного забезпечення для тренування очей. Здійснено опис роботи юзера з програмним забезпеченням для тренування очей. Виконано тестування програмного забезпечення для тренування очей. Результати тестування програмного забезпечення для тренування очей підтвердили, що розроблена програма може використовуватись за призначенням для зняття нервового напруження на органи зору шляхом виконання гімнастики для очей.

5 КЕРІВНИЦТВО ПРОГРАМІСТА

5.1 Призначення й умови застосування програмного забезпечення для тренування очей

Програмний продукт призначений для зняття нервового напруження на органи зору шляхом виконання гімнастики для очей і представляє собою інтерфейс для виконання комплексу вправ.

Програмний продукт для тренування очей повинен надавати можливість одночасного доступу багатьох користувачів, рекомендацій щодо виконання вправ, автоматичного переходу між вправами, що супроводжується звуковим сигналом.

Програма виконує такі функції:

- користувач може за його допомогою виконувати гімнастику для очей;
- наявність таких вправ, як: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- наявність таймеру та анімації для кожної вправи для тренування очей;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до програми для тренування очей багатьох користувачів з пристроїв на різних операційних системах.

Для нормального функціонування програми для тренування очей повинні виконуватися такі умови до апаратного та програмного забезпечення:

- процесор із тактовою частотою 2 GHz та більше;
- наявність IBM-сумісного персонального комп'ютеру;
- оперативна пам'ять 1 ГБ та більше;
- наявність будь-якої з операційних систем Windows, Linux, Mac;

- наявність браузеру для виходу в Інтернет (бажано, Google Chrome);
- вільний простір на жорсткому диску не менше, ніж 15 Мб.

5.2 Характеристики програми

Програма виконана за допомогою мов програмування C# та JavaScript, функції бібліотеки React та інтегрованого середовища розробки Microsoft Visual Studio, а також хмарного сервісу Azure.

Тривалість одного сеансу програми становить 3 хвилини 25 секунд, оскільки вона представляє собою чітку послідовність вправ зі встановленим часом на їх виконання.

Режим роботи – он-лайн, при чому програмою може користуватися одночасно багато користувачів, оскільки програмний продукт має клієнт-серверну архітектуру.

Користувач може помилитися лише ввівши неправильні значення адреси у межах сайту, у такому разі програма за замовченням відкриє стартову сторінку і самовідновиться.

5.3 Звернення до програми

Для запуску програму необхідно підключитись до мережі Інтернет, ввести у рядок браузеру відповідну адресу і, в залежності від швидкості з'єднання почекати декілька секунд. За потреби, натиснути кнопку «Докладніше» для ознайомлення із загальними рекомендаціями.

Для початку виконання гімнастики для очей необхідно натиснути кнопку «Розпочати тренування», далі – «Почати», і впродовж трьох хвилин двадцяти п'яти секунд виконувати зазначені вправи. Після закінчення вправ їх можна повторити, знов натиснувши кнопку «Розпочати тренування», або закрити вкладку браузера.

5.4 Вхідні дані

До вхідних даних програмного забезпечення для тренування очей відносяться запити користувача: натискання кнопки старту або переходу на сторінку із загальними рекомендаціями.

5.5 Вихідні дані

Вихідні дані програмного забезпечення для тренування очей організовані за допомогою інтерфейсу програми для тренування очей, що включає рекомендації до кожної з вправ, таймери та анімаційні елементи.

ВИСНОВКИ

У дипломній кваліфікаційній роботі бакалавра виконано розробку програмного забезпечення для тренування очей. Проаналізовано проблему створення програмних продуктів для тренування очей. Проаналізовано програмне забезпечення для автоматизації процесу рекомендацій щодо виконання вправ для тренування очей. Виявлено, що найбільш популярні на сьогоднішній день сервіси та програми, які дозволяють виконувати тренування для очей, беззаперечно, мають переваги, які полягають у виконанні гімнастики для очей, не відриваючись від комп'ютеру або мобільного телефону. Проте, деякі з відомих сервісів потребують постійного підключення до мережі Інтернет, інші не мають анімаційних елементів, не містять таймеру, а також характеризуються незначною кількістю вправ для тренування очей. Тому за результатом аналізу програм-аналогів було прийнято рішення про доцільність розробки власного програмного забезпечення для тренування очей, а також сформовано основні функціональні вимоги до розроблюваної програми.

Для розробки програмного забезпечення для тренування очей обрано мову C#, оскільки ця мова дозволяє ефективно створювати графічний інтерфейс користувача, є об'єктно-орієнтованою мовою програмування, а також характеризується зручністю використання для створення програмного забезпечення для тренування очей. Також для розробки програмного забезпечення для тренування очей використовується мова JavaScript (JS), яка є динамічною, об'єктно-орієнтованою прототипною мовою програмування.

Для створення програмного забезпечення для тренування очей обрано бібліотеку React мови JavaScript, яка є відкритою JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків.

Проаналізовано середовища розробки програмного забезпечення. Зважаючи на наведену порівняльну характеристику у табл.2.1 та виявлені недоліки таких систем, як SharpDevelop та MonoDevelop, було вирішено обрати саме інтегроване середовище розробки Visual Studio.

Також для розробки програмного забезпечення для тренування очей використано платформу Microsoft Azure, що являє собою хмарну платформу та інфраструктуру корпорації Microsoft, призначену для розробників застосунків хмарних обчислень і покликана спростити процес створення он-лайн додатків.

Визначено загальну структуру задач програмного продукту, що дозволяє розробити програмне забезпечення відповідно до проаналізованих та виявлених вимог. Створено програмний продукт, який повністю реалізує поставлену задачу.

Розроблене програмне забезпечення для тренування очей виконує такі функції:

- можливість виконання користувачем гімнастики для очей;
- наявність таких вправ, як: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- наявність таймеру та анімації для кожної вправи для тренування очей;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до програми для тренування очей багатьох користувачів з пристроїв на різних операційних системах.

Описано основні умови виконання програмного забезпечення для тренування очей. Здійснено опис роботи юзера з програмним забезпеченням для тренування очей. Виконано тестування програмного забезпечення для тренування очей. Результати тестування програмного забезпечення для тренування очей підтвердили, що розроблена програма може

використовуватись за призначенням для зняття нервового напруження на органи зору шляхом виконання гімнастики для очей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ефективні вправи для очей [Електронний ресурс]. – Режим доступу : <https://centeroka.com.ua/uk/publikatsyy/inshi-publikatsii/efektyvni-vpravy-dlia-ochei.html>.
2. Гімнастика для очей. – 7 кращих вправ для оновлення та покращення зору [Електронний ресурс]. – Режим доступу : <http://fit4brain.com/8909>.
3. Гімнастика для очей: найкращі вправи для поліпшення гостроти зору [Електронний ресурс]. – Режим доступу : <http://astravisus.rv.ua/article/gimnastika-dlya-ochej-najkrashhi-vpravi-dlya-polipshennya-gostroty-zoru-34.html>.
4. Вікові норми, гімнастика і корисні продукти для здорових очей [Електронний ресурс]. – Режим доступу : <https://learning.ua/blog/202004/vikovi-normy-himnastyka-i-korysni-produkty-dlia-zdorovykh-ocheniats/>
5. Гімнастика для очей : 10 легких вправ [Електронний ресурс]. – Режим доступу : <https://contactlinza.com.ua/uk/articles/blog/gimnastika-dlya-glaz/>.
6. Топ-5 порад — як зберегти зір працюючи за комп'ютером [Електронний ресурс]. – Режим доступу : <https://oppb.com.ua/news/top-5-porad-yak-zberegty-zir-pracyuyuchy-za-kompyuterom>.
7. Blimb : Зарядка для очей [Електронний ресурс]. – Режим доступу : <http://blimb.su/>.
8. Eyexercise : гімнастика для очей [Електронний ресурс]. – Режим доступу : <http://glaz.freetonik.com/>.
9. Вправи для очей [Електронний ресурс]. – Режим доступу : <https://play.google.com/store/apps/details?id=com.shvagerfm.eyecorrector1&hl=ru>
10. Язык программирования C# [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/C_Sharp.

11. Мова програмування JavaScript [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/JavaScript>.
12. Бібліотека React [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/React>.
13. Керівництво по ASP.NET Core 2.0 [Електронний ресурс]. – Режим доступу : <https://metanit.com/sharp/aspnet5/>.
14. Application Startup in ASP.NET Core 2.0 [Electronic resource]. – Access mode : <https://www.tektutorialshub.com/application-startup-in-asp-net-core/>.
15. Хмарні рішення Microsoft Azure [Електронний ресурс]. – Режим доступу : <http://integritysys.com.ua/solutions/privatecloud-solution-azure/>.

ДОДАТОК А
Технічне завдання

Вступ

Для розв'язання проблеми надмірного навантаження на органи зору при роботі з комп'ютером розроблено програмне забезпечення для тренування очей.

A.1 Підстави для розробки

Робота виконується на основі завдання на дипломну кваліфікаційну роботу бакалавра за темою “Програмне забезпечення для тренування очей” і затверджене наказом № 103 від 30 березня 2021 р. по по Національному університету «Запорізька політехніка».

A.2 Призначення розробки

Програмне забезпечення призначене для розв'язання проблеми надмірного навантаження на органи зору при роботі з комп'ютером розроблено програмне забезпечення для тренування очей. Програмний продукт повинен надавати можливість одночасного доступу багатьох користувачів, рекомендацій щодо виконання вправ, автоматичного переходу між вправами, що супроводжується звуковим сигналом. Проблема зняття значного напруження на очі є актуальною, оскільки на сьогоднішній день інформаційні технології дуже активно розвиваються, і люди все більше часу проводять перед екраном моніторів. Тож виникає потреба в створенні програмного продукту, який би дозволив виконувати гімнастику для очей в інтерактивному режимі з наочним представлення вправ.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Програмний продукт для тренування очей має виконувати такі функції:

- користувач може за його допомогою виконувати гімнастику для очей;
- наявність таких вправ, як: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- наявність таймеру та анімації для кожної вправи;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до ПП багатьох користувачів з пристроїв на різних операційних системах.

До вхідних даних програмного забезпечення для тренування очей відносяться запити користувача: натискання кнопки старту тренування або переходу на сторінку із загальними рекомендаціями.

Вихідні дані програмного забезпечення для тренування очей організовані у вигляді інтерфейсу програми для тренування очей, що включає рекомендації до кожної з вправ, таймер та анімаційні елементи.

Інтерфейс користувача програмного забезпечення для тренування очей повинен забезпечити просту й ефективну взаємодію користувача з програмним продуктом для тренування очей.

А.3.2 Вимоги до надійності

Робота користувача з програмним продуктом для тренування очей повинна спрощуватися шляхом рекомендацій користувачу, як виконувати ту чи іншу вправу.

Безперебійний доступ до програмного продукту забезпечується хмарним сервісом Azure, який відіграє роль серверу у архітектурі програмного продукту.

A.3.3 Вимоги до складу та параметрів технічних засобів

Для відсутності проблем функціонування програмного забезпечення для тренування очей на комп'ютері користувача повинні виконуватися такі умови до апаратної та програмної частини:

- процесор із тактовою частотою 2 GHz та більше;
- наявність IBM-сумісного персонального комп'ютеру;
- оперативна пам'ять 1 ГБ та більше;
- наявність будь-якої з операційних систем Windows, Linux, Mac;
- наявність браузеру для виходу в Інтернет (бажано, Google Chrome);
- вільний простір на жорсткому диску не менше, ніж 15 Мб.

A.3.4 Вимоги до програмної документації

Склад програмної документації програмного забезпечення для тренування очей повинен включати в себе:

- технічне завдання на розробку програмного забезпечення для тренування очей;
- керівництво програміста програмного забезпечення для тренування очей;
- текст програми.

A.3.5 Вимоги до маркування та упакування

Програмне забезпечення для тренування очей повинно мати маркування з позначенням найменування.

Носій з програмним забезпеченням для тренування очей разом з документацією укладається у відповідну тару згідно з вимог до пакування.

A.4 Порядок контролю та приймання

Дипломна кваліфікаційна робота бакалавра повинна виконуватися у повній відповідності з календарним планом, виконання етапів має контролюватися керівником роботи. Останнім етапом є захист роботи перед екзаменаційною комісією.

ДОДАТОК Б
Текст програми

```
// Program.cs
using Microsoft.AspNetCore;
using Microsoft.AspNetCore.Hosting;
using System;

namespace EyeTrainerReact
{
    public class Program
    {
        public static void Main(string[] args)
        {
            BuildWebHost(args).Run();
        }

        public static IWebHost BuildWebHost(string[] args) =>
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build();
    }
}

// Лістинг StartUp.cs
using System;
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.AspNetCore.Http;
using React.AspNet;

namespace EyeTrainerReact
{
    public class Startup
    {
        public
        IServiceProvider
        ConfigureServices(IServiceCollection services)
        {
            services.AddSingleton<IHttpContextAccessor,
HttpContextAccessor>();
            services.AddReact();
            services.AddMvc();
        }
    }
}
```

```

        return services.BuildServiceProvider();
    }

    public void Configure(IApplicationBuilder app)
    {
        app.UseReact(config => { });
        app.UseDefaultFiles();
        app.UseStaticFiles();

        app.UseMvc(routes =>
        {
            routes.MapRoute(
                name: "default",
                template:
"{controller=Home}/{action=Index}/{id?}");

            routes.MapSpaFallbackRoute(
                name: "spa-fallback",
                defaults: new { controller = "Home",
action = "Index" });
        });
    }
}

// Лістинг Index.cshtml
@{
    ViewData["Title"] = "Home Page";
}

<link                href="@Url.Content("~/css/Styles.css")"
rel="stylesheet" type="text/css" />
<div class="stage">
    <div id="content"></div>
    <center><div id="eyetop"></div></center>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/react/15.3.2/react.js">
</script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/react/15.3.2/react-
dom.js"></script>

```

```

    <script src="js/index.jsx"></script>
</div>

```

```
// Лістинг index.jsx
```

```

class EyeTop extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      state0: 0,
      state1: 20,
      state2: 20,
      state3: 10,
      state4: 20,
      state5: 20,
      state6: 10,
      state7: 20,
      state8: 20,
      state9: 10,
      state10: 20,
      state11: 15,
      state12: 15
    }
  }

  componentWillUnmount() {
    clearInterval(this.timer)
  }

  tick() {
    if (this.state.state0 < 12) {
      this.setState({ state0: (this.state.state0 + 1)
    })

    }
    else if (this.state.state1 > 1)
    {
      this.setState({ state1: (this.state.state1 - 1)
    })

    }
    else if (this.state.state1 == 1)
    {
      this.playAudio(this)
    }
  }
}

```

```

        this.setState({ state1: (this.state.state1 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state2 > 1)
    {
        this.setState({ state2: (this.state.state2 - 1)
    })
    }
    else if (this.state.state2 == 1)
    {
        this.playAudio(this)
        this.setState({ state2: (this.state.state2 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state3 > 1)
    {
        this.setState({ state3: (this.state.state3 - 1)
    })
    }
    else if (this.state.state3 == 1)
    {
        this.playAudio(this)
        this.setState({ state3: (this.state.state3 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state4 > 1)
    {
        this.setState({ state4: (this.state.state4 - 1)
    })
    }
    else if (this.state.state4 == 1)
    {
        this.playAudio(this)
        this.setState({ state4: (this.state.state4 - 1)
    })

```

```

        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state5 > 1)
    {
        this.setState({ state5: (this.state.state5 - 1)
    })
    }
    else if (this.state.state5 == 1)
    {
        this.playAudio(this)
        this.setState({ state5: (this.state.state5 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state6 > 1)
    {
        this.setState({ state6: (this.state.state6 - 1)
    })
    }
    else if (this.state.state6 == 1)
    {
        this.playAudio(this)
        this.setState({ state6: (this.state.state6 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }
    else if (this.state.state7 > 1)
    {
        this.setState({ state7: (this.state.state7 - 1)
    })
    }
    else if (this.state.state7 == 1)
    {
        this.playAudio(this)
        this.setState({ state7: (this.state.state7 - 1)
    })
        this.setState({ state0: (this.state.state0 + 1)
    })
    }

```

```

    }
    else if (this.state.state8 > 1)
    {
        this.setState({ state8: (this.state.state8 - 1)
    })

    }
    else if (this.state.state8 == 1)
    {
        this.playAudio(this)
        this.setState({ state8: (this.state.state8 - 1)
    })

        this.setState({ state0: (this.state.state0 + 1)
    })

    }
    else if (this.state.state9 > 1)
    {
        this.setState({ state9: (this.state.state9 - 1)
    })

    }
    else if (this.state.state9 == 1)
    {
        this.playAudio(this)
        this.setState({ state9: (this.state.state9 - 1)
    })

        this.setState({ state0: (this.state.state0 + 1)
    })

    }
    else if (this.state.state10 > 1)
    {
        this.setState({ state10: (this.state.state10 - 1)
    })

    }
    else if (this.state.state10 == 1)
    {
        this.playAudio(this)
        this.setState({ state10: (this.state.state10 - 1)
    })

        this.setState({ state0: (this.state.state0 + 1)
    })

    }
    else if (this.state.state11 > 1)

```

```

        {
            this.setState({ state11: (this.state.state11 - 1)
    ))
        }
    else if (this.state.state11 == 1)
    {
        this.playAudio(this)
        this.setState({ state11: (this.state.state11 - 1)
    ))
        this.setState({ state0: (this.state.state0 + 1)
    ))
    }
    else if (this.state.state12 > 1)
    {
        this.setState({ state12: (this.state.state12 - 1)
    ))
    }
    else if (this.state.state12 == 1)
    {
        this.playAudio(this)
        this.setState({ state12: (this.state.state12 - 1)
    ))

        this.setState({ state0: 0 })
        clearInterval(this.timer)
        this.state = {
            state0: 0,
            state1: 20,
            state2: 20,
            state3: 10,
            state4: 20,
            state5: 20,
            state6: 10,
            state7: 20,
            state8: 20,
            state9: 10,
            state10: 20,
            state11: 15,
            state12: 15
        }
    }
    else clearInterval(this.timer)

```

```

    }

    startTimerEye() {
        clearInterval(this.timer)
        this.timer = setInterval(this.tick.bind(this), 50)
    }

    startTimerStart() {
        clearInterval(this.timer)
        this.timer = setInterval(this.tick.bind(this), 70)
    }

    startTimerBlocks() {
        clearInterval(this.timer)
        this.timer = setInterval(this.tick.bind(this), 1000)
    }

    playAudio() {
        var myAudio = new Audio;
        myAudio.src = "/audio/zvuk.wav";
        myAudio.play();
    }

    render() {
        switch (this.state.state0) {
            case 0:
                return (
                    <div>
                        <div className="eye pos0"> </div>
                        <div>
                            <button
                                onClick={this.startTimerEye.bind(this)}>Розпочати
                                тренування
                            </button> </div>
                            <a
                                href="about.html">
                                Докладніше </a>
                            </div>)
            case 1:
                return (<div className="eye pos1"> </div>)
            case 2:
                return (<div className="eye pos2"> </div>)
            case 3:
                return (<div className="eye pos3"> </div>)
        }
    }

```

```

case 4:
    return (<div className="eye pos4"> </div>)
case 5:
    return (<div className="eye pos5"> </div>)
case 6:
    return (<div className="eye pos6"> </div>)
case 7:
    return (<div className="eye pos7"> </div>)
case 8:
    return (<div className="eye pos8"> </div>)
case 9:
    return (<div className="eye pos9"> </div>)
case 10:
    return (<div className="eye pos10"> </div>)
case 11:
    clearInterval(this.timer)
    return (<div className="instruction_block">
        <h1 className="title">Рекомендації</h1>
        <p>В комплексі набір вправ, під час яких
потрібно дивитися у напрямку, який показує кружечок.</p>
        <p>Вілий кружечок вказує тільки напрямок
- погляд відводьте максимально, але щоб не було напруги в очах.</p>
        <p>У крайніх точках затримуйте погляд на
секунду.</p>

        <p>Після виконання кожної вправи, за
бажанням, можна легко замружитись або покліпати декілька секунд -
це допоможе розслабити очі.</p>
        <p>Тривалість одного заняття: 3 хвилини
25 секунд.</p>

        <button
onClick={this.startTimerStart.bind(this)}> Почати </button >
        </div>)
case 12:
    return (
        <div>
            <div className='blim'> </div>

            <h1
className="title">Розслабтесь</h1>
            <div className='timer'>
                <h1>{this.state.state1}</h1>

```

```

<div>{this.startTimerBlocks(this)}</div>
                                <h4>    Заплющіть    очі    і    не
відкривайте до того, як почуєте звуковий сигнал </h4>
                                </div>
                                </div>
                                )
case 13:
    return (
        <div>
            <div className='updown'> </div>
            <h1 className='title'>Вгору-вниз</h1>
            <div className='timer'>
                <h1>{this.state.state2}</h1>

<div>{this.startTimerBlocks(this)}</div>
                                <h4>    На секунду затримуйте погляд
у верхній та нижній точках </h4>
                                </div>
                                </div>
                                )
case 14:
    return (
        <div>
            <div      className='sqrtcircle'>

</div>
                                <h1      className="title">Покліпайте
очима</h1>
                                <div className='timer'>
                                    <h1>{this.state.state3}</h1>

<div>{this.startTimerBlocks(this)}</div>
                                <h4>    Покліпайте    очима    або
прикрийте долонями очі до сигналу </h4>
                                </div>
                                </div>
                                )
case 15:
    return (
        <div>

```



```

        </div>
    </div>
    )
    case 18:
        return (
            <div>

                <div className='clock'> </div>
                <h1 className="title">За годинниковою
стрількою</h1>

                <div className='timer'>
                    <h1>{this.state.state7}</h1>

<div>{this.startTimerBlocks(this)}</div>
                    <h4> Не поспішайте, виконуйте
плавно</h4>

                </div>
            </div>
        )
    case 19:
        return (
            <div>

                <div className='spiral'> </div>
                <h1 className="title">По спіралі</h1>
                <div className='timer'>
                    <h1>{this.state.state8}</h1>

<div>{this.startTimerBlocks(this)}</div>
                    <h4> Намагайтесь робити не менше,
ніж чотири оберти за раз</h4>

                </div>
            </div>
        )
    case 20:
        return (
            <div>

                <div className='clips2'> </div>
                <h1
                    className="title">Покліпайте
очима</h1>

                <div className='timer'>
                    <h1>{this.state.state9}</h1>

```

```

<div>{this.startTimerBlocks(this)}</div>
                                <h4>      Покліпайте      очима      або
прикрийте очі долонями до сигналу</h4>
                                </div>
                                </div>
                                )
                                case 21:
                                return (
                                <div>
                                <h1      className="title">Далеко-
близько</h1>

                                <div className='timer'>
                                <h1>{this.state.state10}</h1>

                                <div>{this.startTimerBlocks(this)}</div>
                                <h4>      Подивіться      на      предмет      за
межами екрану, потім на екран</h4>
                                </div>
                                </div>
                                )
                                case 22:
                                return (
                                <div>

                                <div className='blim'> </div>

                                <h1
className="title">Розслабтесь</h1>
                                <div className='timer'>
                                <h1>{this.state.state11}</h1>

                                <div>{this.startTimerBlocks(this)}</div>
                                <h4>      Заплющіть очі до сигналу, ми
майже закінчили тренування</h4>
                                </div>
                                </div>
                                )
                                case 23:
                                return (
                                <div>
                                <h1 className="title"> </h1>
                                <h1 className="title"> </h1>

```

```

        <h1 className="title"> Ви молодець!
Тепер трохи відпочиньте від комп'ютера. </h1>

```

```

<div>{this.startTimerBlocks(this)}</div>
        </div>

```

```

        )
    }
}

```

```

function EyeTop() {
    return (
        <div className='eye'>
            </div>
        )
}

```

```

ReactDOM.render(
    <EyeTop />,
    document.getElementById("eyetop")
)

```

```

function App() {
    return (
        <div>
            <h1> <center> </center> </h1>
        </div>
    )
}

```

```

ReactDOM.render(
    <App />,
    document.getElementById("content")
)

```

```

// Styles.ccs
a {
    font-size: 16px;
    color: white;
}

```

```

.eye {
    margin: 50px;
    width: 170px;
    height: 90px;

    background: url(../img/eyes.png) no-repeat;
    align-items: center;
    background-position: 0 0px;
}

.pos0 {
    background-position: 0 0px;
}
.pos1 {
    background-position: 0 -90px;
}
.pos2 {
    background-position: 0 -180px;
}
.pos3 {
    background-position: 0 -270px;
}
.pos4 {
    background-position: 0 -360px;
}
.pos5 {
    background-position: 0 -450px;
}
.pos6 {
    background-position: 0 -540px;
}
.pos7 {
    background-position: 0 -630px;
}
.pos8 {
    background-position: 0 -720px;
}
.pos9 {
    background-position: 0 -810px;
}
.pos10 {

```

```
background-position: 0 -900px;
}
```

```
.timer {
  margin: 2em auto;
  width: 300px;
  height: 300px;
  color: white;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
}
```

```
h1 {
  font-size: 4em;
}
```

```
button {
  font-size: 16px;
  background-color: transparent;
  color: white;
  border: 3px double;
  border-radius: 0.4em;
  margin: 5em;
  padding-left: 2em;
  padding-right: 2em;
  padding-top: 0.5em;
  padding-bottom: 0.5em;
  display: inline-block;
}
```

```
BODY {
  margin: 0px;
}
```

```
.stage {
  position: relative;
  width: 100%;
  height: 662px;
```

```

        overflow: hidden;
        animation: sky 15s infinite linear;
    }

    .leftright {
        position: absolute;
        left: 0;
        width: 2.5em;
        height: 2.5em;
        margin-top: 19em;
        background: white;
        border-radius: 50%;
        box-shadow: 0 0 40px yellow;
        animation: 4s infinite cubic-bezier(1,0,0,1);
        animation-name: leftrightanim;
    }

    .blim {
        position: absolute;
        margin-left: 47.5em;
        margin-top: 19em;
        width: 2.5em;
        height: 2.5em;
        background: white;
        border-radius: 50%;
        box-shadow: 0 0 40px lime;
        animation: 1s infinite;
        animation-name: blimanim;
    }

    .updown {
        position: absolute;
        margin-left: 47.5em;
        width: 2.5em;
        height: 2.5em;
        background: white;
        border-radius: 50%;
        box-shadow: 0 0 40px lime;
        animation: 4s infinite cubic-bezier(1,0,0,1);
        animation-name: updownanim;
    }

```

```
.sqrtcircle {
  position: absolute;
  margin-left: 47.5em;
  margin-top: 19em;
  width: 2.5em;
  height: 2.5em;
  background: white;
  border-radius: 50%;
  box-shadow: 0 0 40px white;
  animation: 11s infinite;
  animation-name: sqrtcircleanim;
}
```

```
.diagonal {
  position: absolute;
  margin-left: 0em;
  margin-top: 37em;
  width: 2.5em;
  height: 2.5em;
  background: white;
  border-radius: 50%;
  box-shadow: 0 0 20px yellow;
  animation: 4s infinite cubic-bezier(1,0,0,1);
  animation-name: diagonalanim;
}
```

```
.clips{
  position: absolute;
  margin-left: 47.5em;
  margin-top: 19em;
  width: 2.5em;
  height: 2.5em;
  background: white;
  border-radius: 50%;
  box-shadow: 0 0 40px white;
  animation: 11s infinite;
  animation-name: clipsanim;
}
```

```
.clips2 {
```

```

    position: absolute;
    margin-left: 47.5em;
    margin-top: 19em;
    width: 2.5em;
    height: 2.5em;
    background: white;
    border-radius: 50%;
    box-shadow: 0 0 40px white;
    animation: 11s infinite;
    animation-name: clips2anim;
}

.clock {
    position: absolute;
    margin-left: 47.5em;
    margin-top: 19em;
    width: 2.5em;
    height: 2.5em;
    background: white;
    border-radius: 50%;
    box-shadow: 0 0 40px red;
    animation: 5s infinite linear;
    animation-name: clockanim;
}

.spiral {
    position: absolute;
    margin-left: 47.5em;
    margin-top: 19em;
    width: 2.5em;
    height: 2.5em;
    background: white;
    border-radius: 50%;
    box-shadow: 0 0 40px red;
    animation: 5s infinite linear;
    animation-name: spiralanim
}

@keyframes sky {
    0%    { background: #666666}
    25%   { background: #6666aa}

```

```

    50% { background: #aa6666}
    75% { background: #66aa66}
    100% { background: #666666}
}

```

```

@keyframes move {
    0% { margin-left: 1em;}
    50% { margin-left: 92em;}
    100% { margin-left: 1em;}
}

```

```

@keyframes updownanim {
    0% { margin-top: 19em;}
    25% { margin-top: -5em;}
    50% { margin-top: 19em;}
    75% { margin-top: 38em;}
    100% { margin-top: 19em;}
}

```

```

@keyframes blimanim {
    50% {
        height: 3.5em;
        width: 3.5em;
        margin-left: 47em;
        margin-top: 18.5em;
    }

    100% {
        height: 2.5em;
        width: 2.5em;
        margin-left: 47.5em;
        margin-top: 19em;
    }
}

```

```

@keyframes sqrtcircleanim {
    0% { box-shadow: 0 0 40px lime;}
    50% { box-shadow: 0 0 40px white;}
    100% { box-shadow: 0 0 40px yellow;}
}

```

```

@keyframes leftrightanim {
    0%    { margin-left: 47.5em;}
    25%   { margin-left: 1em;}
    50%   { margin-left: 47.5em;}
    75%   { margin-left: 93em;}
    100%  { margin-left: 47.5em;}
}

@keyframes diagonalanim {
    0% {
        margin-left: 1em;
        margin-top: 38em;
    }

    25% {
        margin-left: 1em;
        margin-top: 0em;
    }

    50% {
        margin-left: 93em;
        margin-top: 38em;
    }

    75% {
        margin-left: 93em;
        margin-top: 0em;
    }

    100% {
        margin-left: 1em;
        margin-top: 38em;
    }
}

@keyframes clipsanim{
    0%    { box-shadow: 0 0 40px yellow;}
    50%   { box-shadow: 0 0 40px white;}
    100%  { box-shadow: 0 0 40px red;}
}

```

```

@keyframes clips2anim {
    0%    { box-shadow: 0 0 40px red;}
    50%   { box-shadow: 0 0 40px white;}
    100%  { box-shadow: 0 0 40px lime;}
}

@keyframes clockanim {
    from { transform: rotate(0deg) translateX(275px);}
    to   { transform: rotate(360deg) translateX(275px);}
}

@keyframes spiralanim {
    from { transform: rotate(0deg) translate(300px);}
    to   { transform: rotate(-900deg) translate(0px);}
}

.instruction_block {
    max-width: 800px;
    padding: 0 60px;
    margin: 0 auto;
    color: #fff;
    font-size: 16px;
    text-align: left;
}

.title {
    margin-top: 2em;
    color: #fff;
    font-size: 40px;
    letter-spacing: -1px;
    line-height: 35px;
    max-width: 600px;
    text-align: center;
    padding-bottom: 32px;
}

#bgg{
    background-color: mintcream;
}

.pmod{

```

```
margin-left: 10em;
margin-top: 3em;
margin-right: 20em;
margin-bottom: 3em;
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;
font-size: 18px;
}

.nota { color:blue; }

p { font-size: larger;}

h3 { font-size: 3em;}
```

ДОДАТОК В
Слайди презентації

Національний університет “Запорізька політехніка”
кафедра ПЗ

Дипломна кваліфікаційна робота

**Програмне забезпечення
для тренування очей**

Виконав
ст. гр. КНТ-227

О.О. Щербаков

Керівник
професор, к.т.н., доцент

Г.В. Неласа

Рисунок В.1 – Слайд 1

Об’єктом дослідження дипломної роботи є процес комп’ютерної імітації вправ для тренування очей.

Предмет дослідження – програмне забезпечення для тренування очей.

Мета роботи – розроблення програмних засобів для виконання гімнастики для очей.

Рисунок В.2 – Слайд 2

Завдання роботи

Необхідно було вирішити такі завдання:

- проаналізувати предметну область щодо тренування очей за допомогою комп'ютеру;
- обрати мову програмування;
- розробити структурну схему програмного забезпечення;
- розробити функціональну схему програми;
- виконати конструювання програмного забезпечення для тренування очей;
- оформити програмну документацію на розроблену програму для тренування очей;
- протестувати розроблену програму для тренування очей

3

Рисунок В.3 – Слайд 3

Програмні продукти для тренування очей

Можливості	Blimb	Eyeexercise	Упражнения для глаз
Виконання тренування	+	+	+
Таїмер	+	+	–
Рекомендації	+	+	+
Звуковий супровід	+	+	+
Анімаційні елементи	+	–	+
Робота через Інтернет	+	+	–
Клієнт-серверна архітектура	+	+	+
Кросплатформеність	+	+	–
Повний комплекс вправ	–	–	–
Додаткові елементи інтерфейсу	+	+	–

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння	PHP	C#	Java
Засоби для створення графічного інтерфейсу	+/-	+	+/-
Ефективні засоби об'єктно-орієнтованого програмування	+/-	+	+
Можливість роботи з прототипами	-	+	-
Підтримка парадигми процедурного програмування	+	+	-
Зручність використання для створення програмного забезпечення для тренування очей	-	+	+/-
Можливість роботи з базами даних	+	+	+

5

Рисунок В.5 – Слайд 5

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Microsoft VS	Sharp Develop	MonoDevelop
Зручність для розробки програмного забезпечення для тренування очей	+	+	-
Кросплатформність	+	-	+
Підтримка репозиторіїв	+	+	-
Швидкість модернізації розроблюваного програмного забезпечення	+	-	+
Наявність візуального конструктора	+	+	+

6

Рисунок В.6 – Слайд 6

Структурна схема задач програмного продукту

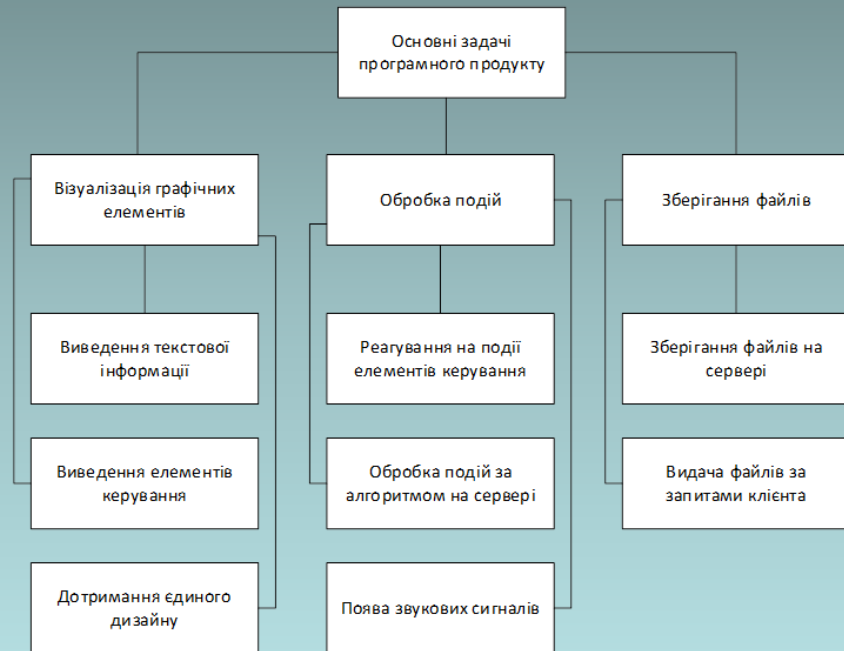


Рисунок В.7 – Слайд 7

Діаграма прецеденту «Тренування»

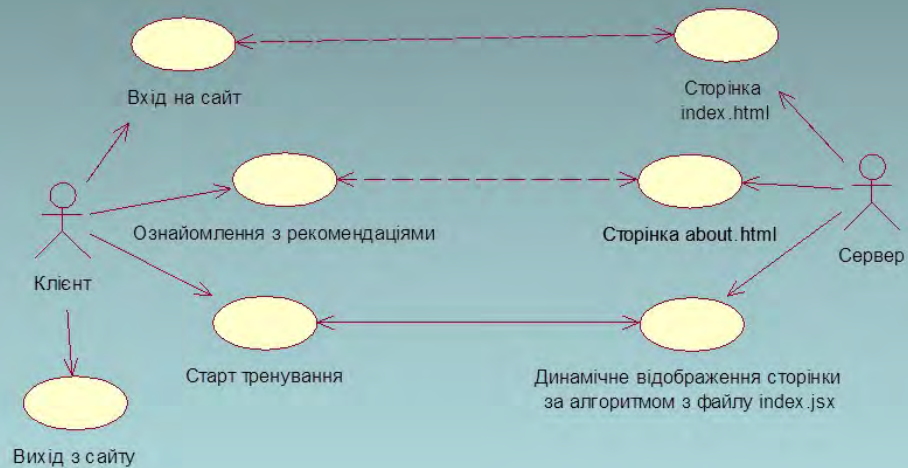


Рисунок В.8 – Слайд 8



Рисунок В.9 – Слайд 9

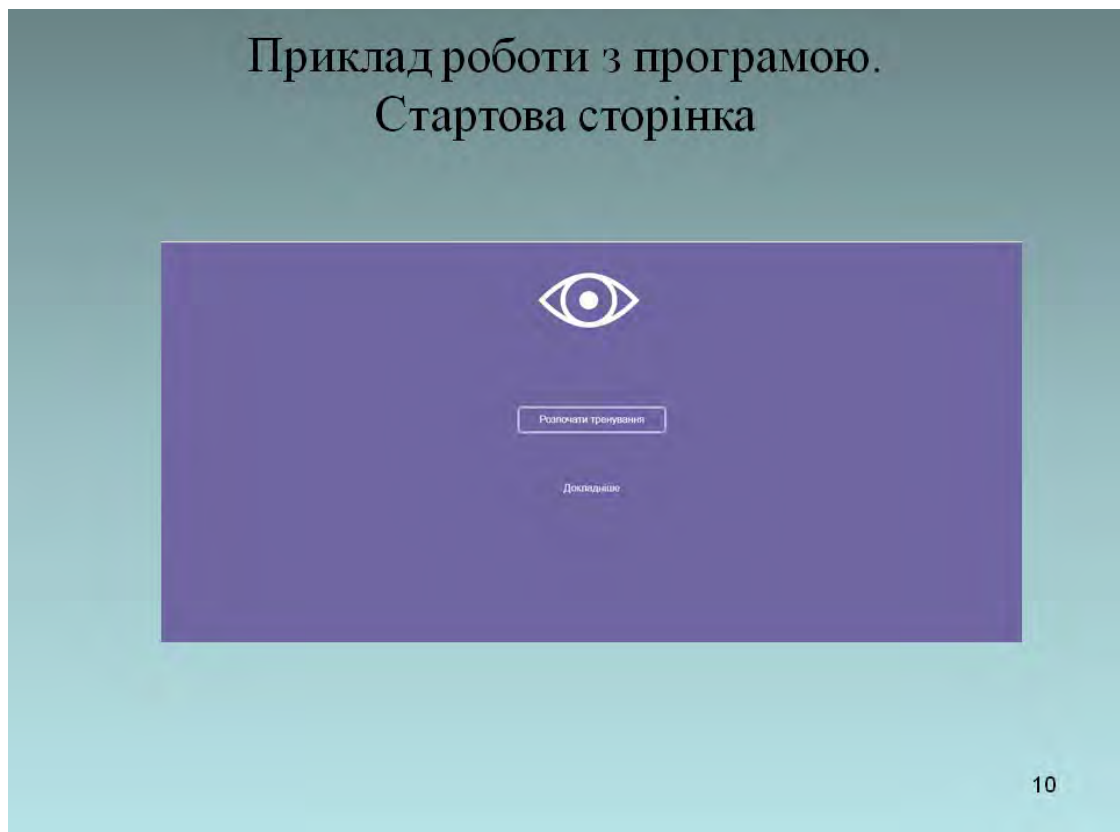


Рисунок В.10 – Слайд 10

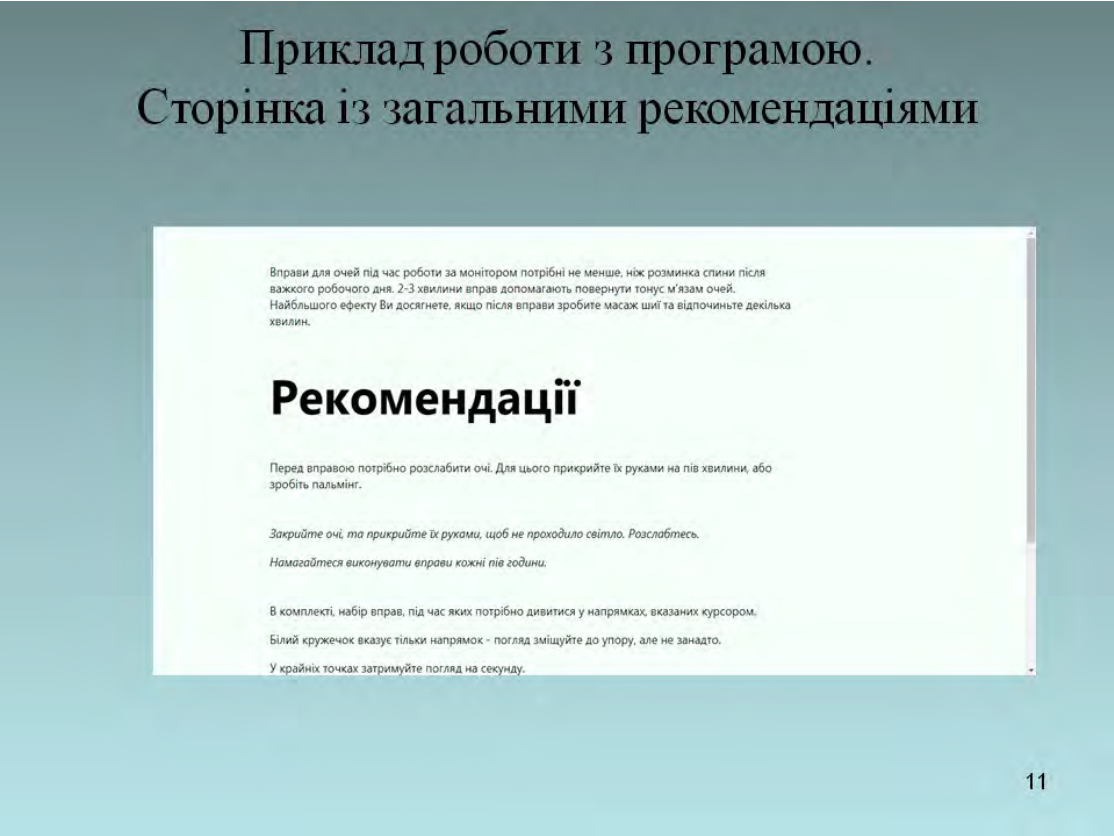


Рисунок В.11 – Слайд 11



Рисунок В.12 – Слайд 12

Висновки

В результаті виконаної роботи було розроблено програмне забезпечення для тренування очей. Розроблене програмне забезпечення забезпечує виконання таких можливостей для користувачів:

- можливість виконання користувачем гімнастики для очей;
- наявність таких вправ, як: «заплющити очі», «вгору-вниз», «покліпати очима», «вправо-вліво», «по діагоналі», «за годинниковою стрілкою», «за спіраллю»;
- наявність таймеру та анімації для кожної вправи для тренування очей;
- автоматичний перехід до наступної вправи, що супроводжується звуковим сигналом;
- одночасний доступ до програми для тренування очей багатьох користувачів з пристроїв на різних операційних системах.

13

Рисунок В.13 – Слайд 13

Висновки

В результаті виконання дипломної кваліфікаційної роботи бакалавра були розв'язані такі завдання:

- аналіз предметної області щодо тренування очей за допомогою комп'ютеру;
- вибір мови програмування;
- розробка структурної схеми програмного забезпечення;
- розробка функціональної схеми програмного забезпечення;
- конструювання програмного забезпечення для тренування очей;
- оформлення програмної документації на розроблену програму для тренування очей;
- тестування розробленої програми для тренування очей.

Проведені випробування показали, що програмне забезпечення для тренування очей повністю відповідає технічному завданню й готове до практичного застосування.

14

Рисунок В.14 – Слайд 14