

Міністерство освіти і науки, молоді та спорту України  
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Кафедра комп'ютерних наук

(повна назва кафедри)

## КУРСОВИЙ ПРОЕКТ

з Об'єктно-орієнтованого програмування

(назва дисципліни)

на тему: «Створення web-додатку Netflix 2.0»

Студента 2 курсу, групи СН-22  
напряму підготовки 6.050101

комп'ютерні науки  
спеціальності 7.05010101 - інформаційні  
управляючі системи та технології

Палки О. В.

(прізвище та ініціали)

Керівник: к.т.н., доцент Бревус В. М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Оцінка за національною шкалою

Кількість балів:            Оцінка ECTS           

Члени комісії:

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

Кафедра Комп'ютерних наук

Дисципліна «Об'єктно-орієнтоване програмування»

Напрямок підготовки 6.050101 «Комп'ютерні науки»

Курс 2 Група СН-22 Семестр IV

## ЗАВДАННЯ

### на курсовий проект

Студентові Палці Олегу Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Створення web-додатку Netflix 2.0

2. Термін здачі студентом закінченої роботи 10.06.2016

3. Вихідні дані до роботи 3.1 Підстава для розробки – виконання курсової роботи.

3.2 Призначення розробки – створити web-додаток Netflix 2.0.

3.3 Вхідні дані – дії користувачів 3.4 Вихідні дані – рейтинг відеоматеріалу та його перегляд.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) \_\_\_\_\_

Вступ. 1 Аналіз технічного завдання. 2 Обґрунтування алгоритму і структури програми. 3 Розробка програми. 4 Тестування програми і результат її виконання.

Висновки. Список використаної літератури. Додатки.

5. Перелік графічного (ілюстративного) матеріалу, якщо передбачено презентація

6. Дата видачі завдання 04.03.2016

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Примітка
1	Технічне завдання – Постановка задачі, визначення вимог, структури даних, методу рішення, тощо	15.04.2016	<b>Виконано</b>
2	Технічний проект - Розробка алгоритму, визначення форми представлення даних, структури програми	25.04.2016	<b>Виконана</b>
3	Робочий проект - Програмування. Розробка документації. Випробування програми.	28.04.2016	<b>Виконано</b>
4	Оформлення пояснювальної записки	04.05.2016	<b>Виконано</b>
5	Представлення роботи до попереднього розгляду	03.06.2016	<b>Виконано</b>
6	Захист роботи на кафедрі	10.06.2016	<b>Виконано</b>

Студент

\_\_\_\_\_

(підпис)

Палка Олег Вікторович

\_\_\_\_\_

(прізвище, ім'я, по батькові)

Керівник роботи

\_\_\_\_\_

(підпис)

к.т.н., доцент Бревус Віталій Миколайович

\_\_\_\_\_

(вчений ступінь, посада, прізвище, ім'я, по батькові)

## ЗМІСТ

АНОТАЦІЯ.....	5
ВСТУП.....	6
РОЗДІЛ І АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	8
РОЗДІЛ ІІ ОБГРУНТУВАННЯ АЛГОРИТМУ І СТРУКТУРИ ПРОГРАМИ.....	26
РОЗДІЛ ІІІ РОЗРОБКА ПРОГРАМИ.....	29
РОЗДІЛ ІV ТЕСТУВАННЯ ПРОГРАМИ І РЕЗУЛЬТАТИ ІІ ВИКОНАННЯ.....	32
ВИСНОВКИ.....	37
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТОК А.....	39

					ТНТУ КПКН 16.027.27.000 ПЗ			
Зм.	Арк	докум.	Підп.	Дата				
Розроб.		Палка О. В.			ЗМІСТ	Літ.	Арк.	Аркушів
Перевір.		Бревус В. М.					4	39
						ТНТУ, ФІС, СН-22		
Зав. каф.								

## АНОТАЦІЯ

Описано об'єктно-орієнтований підхід розробки програмних продуктів мовою програмування C++, а саме – створення web-додатків, що зумовлене використанням фреймворку, і застосовано уніфіковану мову моделювання (UML) для створення абстрактної моделі системи. Результатом роботи є web-додаток, призначений для оцінки та перегляну відеоматеріалів, а також для рекомендації користувачу фільмів, залежно від вподобань інших користувачів за допомогою методу інверсії.

					ТНТУ КПКН 16.027.27.000 ПЗ			
Зм.	Арк	докум.	Підп.	Дата				
Розроб.		Палка О. В.			АНОТАЦІЯ	Літ.	Арк.	Аркушів
Перевір.		Бревус В. М.					5	39
						ТНТУ, ФІС, СН-22		
Зав. каф.								

## ВСТУП

Курсовий проект є комплексним завданням по застосуванню та закріпленню вивченого матеріалу за курс з ведучої дисципліни (в даному випадку з об'єктно-орієнтованого програмування). Він включає в себе розробку повноцінної системи (згідно з варіантом) і дає можливість проявити креативність і свої здібності в програмуванні та на кодовій частині проекту.

Розробляючи проект дуже важливим аспектом є створення раціональної системи та її структури, які б підходили під усі вимоги, та найголовніше – виконувала свою задачу якомога найкраще. У даному курсовому проекті представлена програма web-додатку Netflix 2.0.

Програмування – це процес написання, тестування і підтримки комп'ютерних програм. З розвитком програмування виникла ідея поєднати в межах однієї сутності дані і код, що безпосередньо опрацьовує ці дані. Така сутність отримала назву об'єкт, а відповідний підхід до створення програм називають об'єктно-орієнтованим програмуванням.

Об'єктно-орієнтоване програмування (ООП) – це парадигма програмування, яка розглядає програму як сукупність гнучко пов'язаних між собою об'єктів.

Основні переваги концепції ООП:

- можливість створювати користувацькі типи даних (класи);
- приховування деталей реалізації (інкапсуляція);
- можливість повторного використання коду (наслідування);

Метою даної роботи є створення веб-додатку Netflix 2.0. Для цього використовувалось середовище розробки – Microsoft Visual Studio 2012. Крім цього також використовувався: фреймворк Wt із відповідними бібліотеками,

					ТНТУ КПКН 16.027.27.000 ПЗ			
Зм.	Арк	докум.	Підп.	Дата				
Розроб.		Палка О. В.			ВСТУП		Літ.	Арк.
Перевір.		Бревус В. М.						6
							39	
							ТНТУ, ФІС, СН-22	
Зав. каф.								

фреймворк Bootstrap 3, пакет бібліотек Boost, програма для розробки графічного інтерфейсу і запуску сервера – Wt Designer, утиліта для автоматичного створення програми із вихідного коду CMake.

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення 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 Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Інструменти Visual Studio 2012 допоможуть не тільки в створенні звичних програм для мобільних телефонів і персональних комп'ютерів, але в розробці хмарних застосунків. При цьому процес тестування, зневадження і розгортання програм в «хмарі» аналогічний створенню .NET-застосунків.

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						7
Зм.	Арк.	докум.	Підп.	Дата		

## РОЗДІЛ І

### АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Дана програма створена як веб-додаток Netflix 2.0. Середовище розробки програми: Microsoft Visual Studio 2012. Програма написана на мові програмування C++, вона високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. C++ додає до C об'єктно-орієнтовані можливості. Вона вводить класи, які забезпечують три найважливіші властивості ООП: інкапсуляцію, успадкування і поліморфізм.

Інкапсуляція – це механізм в програмуванні, який пов'язує в одне ціле функції і дані, якими вони маніпулюють, а також захищає їх від зовнішнього доступу і неправильного застосування.

Успадкування – це властивість, з допомогою якої один об'єкт може набувати властивостей іншого. При цьому підтримується концепція ієрархічної класифікації.

Поліморфізм дозволяє писати більш абстрактні програми і підвищити коефіцієнт повторного використання коду.

Веб-додаток повинен:

- передбачити наявність кількох користувачів;
- наявність кількох категорій та фільмів, у кожній із них;
- можливість перегляду фільму;
- можливість оцінити фільм;
- дізнатись користувача із схожими вподобаннями.

Для того, щоб створити даний веб-додаток необхідними, крім середовища розробки ПЗ, є:

- Wt-фреймворк (набір відповідних бібліотек);

					ТНТУ КПКН 16.027.27.000 ПЗ								
Зм.	Арк	докум.	Підп.	Дата									
Розроб.		Палка О. В.			РОЗДІЛ І			Літ.		Арк.		Аркушів	
Перевір.		Бревус В. М.								8		39	
								ТНТУ, ФІС, СН-22					
Зав. каф.													



- Фреймворк Bootstrap 3;
- Boost;
- WtDesigner;
- CMake.

Спершу розглянемо *C++ Web toolkit framework*.

**Wt (Web toolkit)** – це бібліотека C++ для розробки web-додатків API, центральне місце в якому займають віджети, використовує добре перевірені патерни розробки GUI, адаптовані для web. Це дозволяє розробнику абстрагуватись від специфічних для web деталей реалізації, включаючи протоколи взаємозв'язку клієнта і сервера, керування подіями, підтримку графіки, поступову деградацію або прогресивне розширення, а також роботу з URL.

На відміну від багатьох сторінко-орієнтованих фреймворків, Wt була розроблена для створення web-додатків, які подоліють статком, котрі, до того ж, дуже інтерактивні (завдяки задіянню всіх можливостей таких технологій, як WebSockets і Ajax), і доступні (завдяки підтримці браузерів простого HTML), автоматично здійснюючи поступову деградацію або прогресивне розширення. Те, що істинно і просто зробити за допомогою Wt, може зажадати надмірних зусиль при спробі здійснити це самостійно, наприклад: анімаційна зміна віджетів без шкоди для правильної індексації по URL пошуковими роботами чи постійно відкритий відмет чату, який працює навіть в таких старих браузерах Internet Explorer 6.

В бібліотеку входить сервер додатків, які діють як самостійний сервер Http(s)/WebSockets чи інтегруються через FastCGI з іншими web-серверами.

Інтерактивний, безпечний, доступний

Зазвичай, сторінко-орієнтовані фреймворки не намагаються абстрагуватись від лежачих в їх основі технології, серед яких можуть бути HTML/XHTML, JavaScript, CSS, Ajax, WebSockets, Comet, Forms, DHTML, SVG/VML/Canvas). Як наслідок, розробник повинен бути знайомий зі всіма цими технологіями, що постійно розвиваються, а також передбачати по-

ступову деградацію на випадок обмежених можливостей браузера. Структура багатьох web-додатків по цей день в багатьох випадках слідує сторінково-орієнтованій парадигмі часів ранніх версій HTML. Це означає не тільки необхідність реалізувати контролер для обліку переміщення користувача від сторінки до сторінки, але й при використанні Ajax чи WebSockets, - самостійно проектувати і реалізовувати взаємодію клієнта і сервера.

З іншого боку, використання “чистих” Ajax-фреймворків приводить до виснажливого програмування на JavaScript, де необхідно мати діло з причудами браузерів і реалізовувати безпечну взаємодію з ресурсами сервера. Зазвичай, такі додатки не відповідають рекомендаціям принципів доступності web-вмісту різними способами і не можуть бути проіндексовані роботами.

Генерація коду HTML чи використання шаблонів HTML провокує проблеми безпеки, такі як XSS(Cross-Site-Scripting – “Міжсайтовий скриптинг”), ненавмисно дозволяючи коду JavaScript бути встановленому на сторінку, і CSRF(Cross-Site Request Forgery – “Підrobка міжсайтових запитів”) довіряючи інформації про аутентифікацію, що знаходиться в куках(cookies). Згадані проблеми тяжко уникнути при використанні традиційних фреймворків, коли Ви, як розробник, повинні реалізовувати деякий функціонал на JavaScript, який в цьому випадку не зможе бути відфільтрований фреймворком.

Навпроти, web-додаток на базі Wt розробляється тільки з використанням C++ API і бібліотека сама генерує необхідний код HTML, CSS, JavaScript, CGI, SVG/VML/Canvas і Ajax. Безпека і крос-браузерність web-додатків забезпечується Wt. Наприклад, коли це можливо, Wt максимально задіює JavaScript, Ajax і навіть WebSockets, хоча в той же час, Wt-додатки коректно функціонують і без JavaScript. Спочатку Wt запускає простий додаток HTML/CGI, а потім прогресивно розширює його до Ajax-додатку, якщо це виявляється можливим. Wt також забезпечує вбудовану за замовчуванням безпеку.

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						10
Зм.	Арк.	докум.	Підп.	Дата		

### Типові сценарії використання:

- *Високопродуктивні, складні web-додатки, які повністю персоналізовані (і, як наслідок, кешування для яких не приносить вигоди), з повною підтримкою Ajax, і, в то же час, повністю доступні і оптимізовані для пошукових систем.*
- *GUI на базі web для вбудованих систем, завдяки малому розміру web-сервера додатку C++.*
- *GUI на базі web, якому потрібна інтеграція з (існуючими) бібліотеками C++, наприклад, для наукових і інженерних додатків, чи з існуючими настільними додатками C++.*

### Ключові можливості бібліотеки:

- Підтримка основних браузерів, а також браузерів простого HTML.
- Розробка і розсорткування на Unix/GNU Linux чи Microsoft Windows (Visual Studio).
- Ідентична поведінка як при підтримці JavaScript чи Ajax, так і без такої, наскільки це можливо, за допомогою поступової деградації чи прогресивного розширення.
- Інтегрована підтримка Юнікода і локалізації.
- Ефективний рендерінг і низька латентність.
- Підтримка переміщення за допомогою історії браузера, лаконічні URL при доступності історії HTML5, оптимізація для пошуковиків і одноманітна поведінка сесії простого HTML чи Ajax.
- Налаштовувані опції відстежування сесій, або посередництвом перезапису URL, або з допомогою використання кук.
- Висока продуктивність, яка дозволяє розгортати додатки на вбудованих пристроях класу low-end чи енерго-, місце- і бюджетно-зберігаючі розсорткування web-сайтів в Інтернет чи в екстранет.
- Повністю подіє-керований асинхронний ввід/вивід: сесії не прив'язані до потоків і відкриті з'єднання не блокують їх. Навпро-

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						11
Зм.	Арк.	докум.	Підп.	Дата		

ти, потоки необхідні як раз для покращення обслуговування конкурентних запитів і для повторних циклів подій.

Інші вигоди від використання Wt:

- Розробка web-додатків з використанням добре знайомих патернів розробки GUI настільних систем.
- Наявність обширного набору віджетів, які можуть працювати без JavaScript (Але з ним працюють ще краще).
- Єдина специфікація клієнтської і серверної валідації і керування подіями.
- Можливість використання XHTML і CSS для компоновки і оформлення.
- Генерація, яка є відповідною стандартам HTML чи XHTML коду.
- Переносна, згладжена графіка, оптимізована для використання в web (за допомогою вбудованого VML, SVG, HTML5 canvas чи зображень PNG), яка також може бути використана для виводу з PDF.
- Дозволяє уникнути поширених проблем безпеки, оскільки володіє повним контролем над рівнем представлення і про активно фільтрує активні теги і атрибути, не впливаючи на бізнес-логіку, і спрощує аутентифікацію завдяки наявності станів у додатків.
- Надшвидка загрузка і дбайливе використання каналу зв'язку, на які впливає тільки складність картинки, а не розмір додатку. Wt реалізує всі відомі прийоми оптимізації чуйності додатків, використовуючи для цього навіть специфічні особливості різних браузерів.
- Простий API з хорошою крос-браузерною реалізацією ініційованих сервером подій, тобто server push (за допомогою comet чи WebSockets).
- Можливість використання вбудованого сервера httpd для простоти розробки. розсортювання, чи FastCGI/ISAPI для розсортювання на діючих web-серверах.

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						12
Зм.	Арк.	докум.	Підп.	Дата		

Наступним для ознайомлення є **Boost**.

**Boost** – набір бібліотек, які розширюють функціональність C++. Більшість бібліотек поширюються під ліцензією Boost Software License, розробленої для використання як з проектами з відкритим сирцевим кодом, так і закритим. Проект був створений після прийняття стандарту C++, коли багато хто був незадоволеним не включенням в стандарт деяких бібліотек. Багато з фундаторів Boost є членами комітету зі стандартизації C++ і декілька Boost бібліотек були прийняті для включення до Technical Report 1 та C++0x.

Boost має помітну спрямованість на дослідження і розширюваність (мета програмування і узагальнене програмування з активним використанням шаблонів). Завдяки ретельному підбору і контролю якості бібліотеки, включені в Boost, мають високу надійність і продуктивність. Думки щодо використання різняться. Деякі вважають його стандартом де-факто і необхідним доповненням до STL. Деякі, навпаки, уникають всякого використання бібліотеки в проектах, оскільки це зайва залежність в проекті і для використання цих бібліотек програмістові необхідно добре знати C++, оскільки деякі частини Boost вимагають досить хорошої підготовки програміста і є вельми складними.

Бібліотеки які входять у набір Boost:

- Алгоритми.
- Обхід помилок в компіляторах не відповідних стандарту.
- Багатопоточне програмування (multi-threading) або багато потоковість.
- Контейнери.
- Юніт-тестування.
- Структури даних.
- Функціональні об'єкти.
- Узагальнене програмування.
- Графи.
- Введення/виведення.

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						13
Зм.	Арк.	докум.	Підп.	Дата		

- Міжмовна підтримка.
- Літератори.
- Математичні і чисельні алгоритми.
- Робота з пам'яттю.
- Синтаксичний і лексичний розбір.
- «Розумні вказівники».
- Обробка рядків і тексту.
- Мета програмування на основі препроцесора.
- Мета програмування на основі шаблонів.

Для створення проекту веб-додатку із використанням фреймворку Wt у Visual Studio, відповідно на платформі Windows використовуємо версію boost.1.61.0. Для встановлення цієї версії на платформі Windows потрібно перейти на сайт: <https://sourceforge.net/projects/boost/files/boost-binaries> . Тоді вибрати потрібну версію, яка підходить для вашого комп'ютера (на рис. 1.1). В нашому випадку це 64-х розрядна система і Visual Studio 12.

Home / boost-binaries / 1.61.0









Name ▾	Modified ▾	Size ▾	Downloads / Week ▾
↑ Parent folder			
SHA256SUMS.asc	2016-05-13	2.4 kB	15  
SHA256SUMS	2016-05-13	1.5 kB	27  
boost_1_61_0-msvc-14.0-64.exe	2016-05-08	283.4 MB	842  
boost_1_61_0-msvc-12.0-64.exe	2016-05-08	275.7 MB	324  

Рисунок 1.1 – Директорія завантажування Boost

У нашій курсовій роботі ми використаємо такі бібліотеки з набору бібліотек BOOST : thread, system, signals, regex та інші.

Детальніше розглянемо **Boost.Thread**.

Boost.Thread дозволяє C++ програмістам реалізовувати різноманітні, асинхронні, незалежні потоки виконання. Кожний потік має свій особистий

машинний стан, включаючи лічильник програмних кострукцій і реєстри. Програми, які реалізовані як різні потоки називають багатопотоковими програмами, щоб відрізнити їх від традиційних однопоточних програм.

Багатопоточність дає деякі переваги:

- Програми які хочуть продовжувати відповідати на запити, при очікуванні зовнішніх подій, можуть помістити блокуючі операції в окремий потік.
- Добре спроектовані багато потокові програми можуть працювати швидше ніж одно поточні на багатопроцесорних системах. Але погано спроектовані багато потокові програми часто більш повільніші за одно поточні.
- Проектування деяких програм може бути простіше, ніж використання багато поточності. В кінці кінців, наш світ асинхронний.

#### *Основний аналіз*

Крім помилок, які можуть бути в одно поточній програмі, багато потокові програми можуть містити додаткові помилки:

- Стан гонок (Race conditions).
- Глухий кут (Deadlock) (так зване «смертельні обійми» («deadly embrace»)).
- Проблема пріоритетів (Priority failures).

Кожна багато потокова програма повинна проектуватись акуратно, щоб уникнути цих помилок. Це не рідкісні чи екзотичні помилки – вони фактично гарантовано появляться, поки код не буде спроектований належним чином. Проблема пріоритетів буває рідше – але вона не менш серйозна.

Погляд на тестування і відладку. Багатопотокові програми не детерміновані. Іншими словами, одна і та сама програма з тим самим кодом може виконуватись кожний раз по різному. Це може зробити тестування і налагодження надзвичайно складною:

- Помилки часто не повторюються
- Ефект налагодження може сильно скривлювати результати, отримані в цьому варіанті.

- Дебагер повинен підтримувати відображений стан потоків.
- Тестування на одно процесорній системі може не виявити серйозних помилок, які можуть статись при використанні багатопроцесорних систем. В тести потрібно включати різну кількість процесорів.
- Тестування на одно процесорній системі може не виявити серйозних помилок, які можуть статись при використанні багатопроцесорних систем. В тести потрібно включати різну кількість процесорів.
- Для програм які створюють різну кількість потоків для роботи, тести які не проходять всі можливі варіанти можуть пропустити серйозні помилки.

Щодо **Boost.Signals**, то часто при проектуванні систем виникають задачі наступного характеру. В системі появляється сутність, яка являє собою генератор деяких подій. І є цілий ряд інших сутностей, які повинні повідомляти при виникненні цих подій. Найпростіший приклад – на деякій інтерфейсній формі є кнопка, при натисканні якої потрібно виконати певний код. Кнопка – джерело поді. Клас форми – одержувач повідомлення про виникнення події і його обробник. Безпосередньої підтримки даного механізму підписки-повідомлення на мові C++ немає. Але реалізацію цього механізму можна знайти в boost. А саме в бібліотеці boost::signal. У лістингу 1.1 наведено приклад класу джерела події.

Лістинг 1.1 – Приклад класу джерела події

```
#include <iostream>
#include <boost/signal.hpp>
#include <boost/bind.hpp>

////////////////////////////////////
//
// Клас кнопки
//Приклад класу-джерела події
////////////////////////////////////
class SampleButton
{
public:
    // Метод, який додає новий обробник події
    // h - функтор, реалізуючий обробник
    void SetOnClick(boost::function<void()> h)
    {
```



```
private:
```

ТНТУ КПКН 16.027.27.000 ПЗ

Клас `boost::signal` реалізує концепцію функціонально об'єкту, по цьому для ініціалізації події, як сказано вище, достатньо застосувати до цього класу оператор виклику функції. При цьому аргументи, які передаються при виклику, повинні відповідати сигнатурі, переданій в якості параметрів шаблону при оголошенні відповідного члена даних.

Як видно з прикладу, на одну подію можна призначити зразу декілька обробників. При цьому ( в простішому випадку) будуть викликані всі назначені обробники. Правда, послідовність їх виклику розробниками бібліотеки не гарантується. Нерідко для позначення сутності, які описують подію ( члена даних `m_OnClick`) використовують термі *сигнал* (signal). А для позначення підписників на цю подію – термін *слот* (slot).

Власне, тепер ми розглянемо програмний засіб для створення дизайн та розмітки веб-сторінки.

**WtDesigner** – це програма для створення web-додатків на мові C++ на основі бібліотеки Wt. Вона допомагає створювати складні макети програм, які можуть підлаштовуватись автоматично до вашого браузера. Кожен віджет всередині макера легко налаштовується, використовуючи сторінки властивостей які адаптуються до типу редагованого віджету. Також у WtDesigner існує режим попереднього перегляду і режим масштабування для детального редагування. На додаток до основного дизайну, діалогові вікна і впливаючі меню також можуть бути створені. На рисунку 1.2 наведено робочу область.

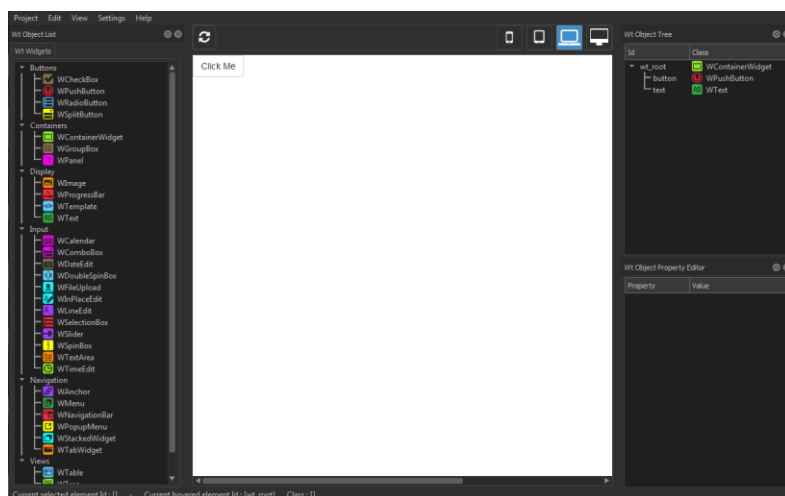


Рисунок 1.2 – Робоча область у WtDesigner

При збереженні, конструкція буде генерувати клас заголовка C++ який містить всі елементи дизайну. Коли створиться цей клас, він буде виробляти об'єкт віджета, який можна прикріпити до вікна браузера або ж до будь якого іншого Wt контейнерного віджета.

Для генерування усіх, необхідних для генерування проекту файлів перш за все необхідно, щоб у програмі Wt Designer був вказаний вірний шлях до Wt-бібліотек у вкладці CMake (на рис.1.3).

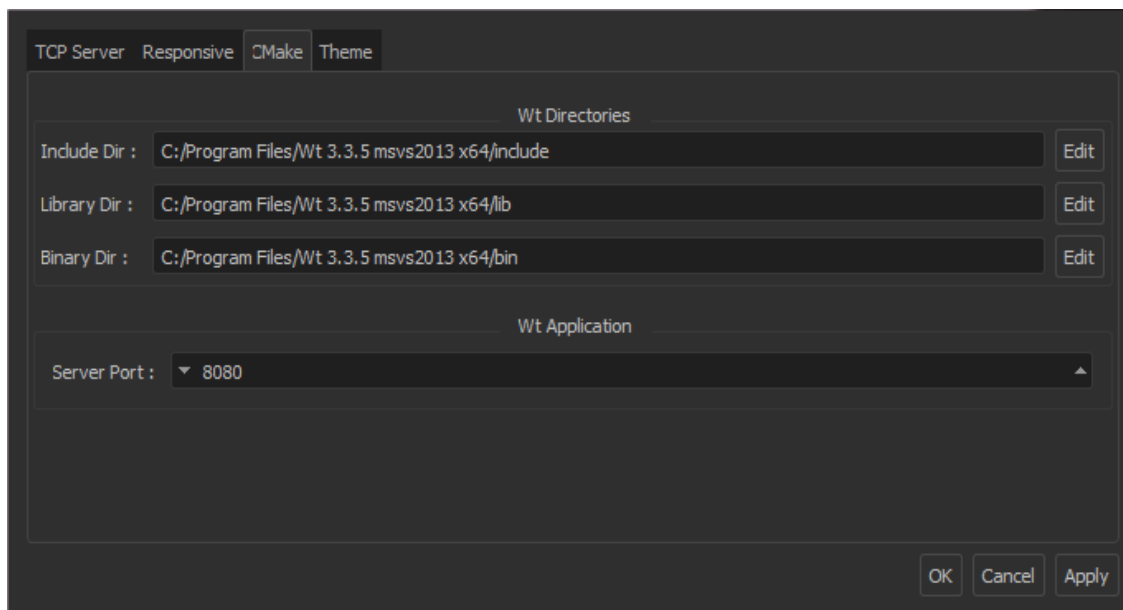


Рисунок 1.3 – Налаштування директорій до Wt

Наступним етапом є, власне, експорт проекту із WtDesigner у файли необхідні для подальшого перетворення у проект Visual Studio. Для цього у меню WtDesigner виконуємо наступні дії: Project → Export to CMake (на рис. 1.4). Після цього буде відображено повідомлення про перебіг експорту – успіх або невдача. У разі успіху CMake-файли, в тому числі і, найважливіший, CMakeLists.txt будуть розташовані у каталозі із відповідним проектом.

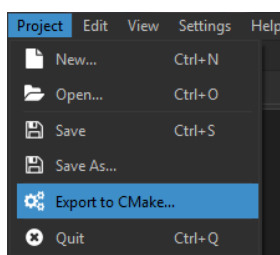


Рисунок 1.4 – Експорт проекту в CMake

Важливим етапом є процес розмітки сторінки, який забезпечується фреймворком **Bootstrap 3**.

**Bootstrap** — це безкоштовний набір інструментів з відкритим сирцевим кодом, призначений для створення веб-сайтів та веб-застосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-застосунків.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій з даним фреймворком є одним з найбільш популярних на GitHub. Серед інших, його використовують NASA і MSNBC

Bootstrap (початкова назва — **Twitter Blueprint**) був розроблений Марком Отто та Джейкобом Торнтоном (Twitter) у якості фреймворку для забезпечення однаковості внутрішніх інструментів Twitter. До появи Bootstrap у розробці інтерфейсу застосовувалися різні бібліотеки, що призводило до появи суперечностей та ускладнювало супровід. За словами Марка Отто, «Маленька група розробників, до складу якої входив і я, спроектувала й побудувала новий внутрішній інструмент, що мав потенціал для створення ширшого рішення. За кілька місяців з'явилася початкова версія Bootstrap, яка являла собою засіб для документування та поширення загальних шаблонів і засобів проектування всередині компанії».

Через кілька місяців до розробки рішення долучилося багато розробників компанії Twitter. Проект було перейменовано з Twitter Blueprint на Bootstrap. Реліз із відкритим сирцевим кодом вийшов 19 серпня 2011 року. Нині проект підтримується невеликою групою розробників на чолі з Марком Отто та Джейкобом Торнтоном, а також широкою спільнотою прихильників.

Bootstrap сумісний з останніми версіями браузерів Google Chrome, Firefox, Internet Explorer, Opera і Safari (деякі з цих браузерів підтримуються не на всіх платформах).

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						20
Зм.	Арк.	докум.	Підп.	Дата		

Bootstrap має модульну структуру і складається переважно з наборів таблиць стилів LESS, які реалізують різні компоненти цього набору інструментів. Розробники можуть самостійно налаштовувати файли Bootstrap, обираючи компоненти для свого проекту.

Основні інструменти Bootstrap:

- **Сітки** (grid) — наперед задані, готові до використання колонки
- **Шаблони** (template) — фіксовані чи адаптивні шаблони сторінок
- **Типографіка** (typography) — опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо
- **Мультимедіа** (media) — засоби управління зображеннями та відео
- **Таблиці** (table) — засоби оформлення таблиць, які зокрема забезпечують сортування
- **Форми** (form) — класи для оформлення як форм, так і деяких подій
- **Навігація** (nav, navbar) — класи для оформлення вкладок, сторінок, меню і панелей навігації
- **Сповіщення** (alert) — класи для оформлення діалогових вікон, підказок і спливаючих вікон
- **Іконочний шрифт** (icon font) — набір іконок у вигляді шрифту, складається майже з 500 компонентів.

Після цієї виконання цієї процедури, ми переходимо до програми **СMake**.

**СMake** – це крос-платформна (cross-platform make) утиліта для автоматичної збірки програми із вихідного коду. При цьому розробник описує параметри в простому форматі у вигляді текстових файлів. Такий опис не залежить від компіляторів, а процес збірки проходить в поєднанні із нативним середовищем, наприклад, Make, Apple's Xcode або Microsoft Visual Studio. Вона дозволяє створювати проекти для CodeBlocks, Eclipse, KDevelop3, MS VC++ і Xcode.

Найважливішою її перевагою є те, що СMake є по-справжньому крос-платформним генератором проектів, що дозволяє створювати єдині описи

проектів для Windows, Linux та інших Unix-систем (включаючи Mac OS X). Зупинимося на цьому детальніше. Важлива відмінність Windows від Linux (з точки зору розробника) полягає в тому, що на платформі Windows немає єдиного стандарту складальних файлів. MS Visual Studio використовує свої файли проектів, C++ Builder - свої, MinGW - свої. Перевага CMake в тому, що ця система здатна генерувати «рідні» складальні файли для всіх перерахованих засобів розробки (як і для багатьох інших).

Важливою особливістю є те, що CMake володіє інтелектуальною системою пошуку інструментів побудови та бібліотек на конкретній платформі (інтроспекція) і автоматичного конфігурування. Завдяки цьому система CMake сама встановлює багато параметрів складальних файлів, які в інших системах управління збіркою доводиться встановлювати вручну. Наприклад, в ОС Linux CMake сам знайде директорію, в якій ви встановили потрібний програмі набір віджетів, а під Windows вам не доведеться вказувати CMake, де встановлені Visual Studio або C++ Builder. Все це робить пакет CMake дуже простим і зручним у використанні.

Алгоритм роботи CMake є наступним:

- 1) Для того, щоб скласти проект програми створюється файл CMakeLists.txt, в якому описуються параметри збірки (розташування файлів вихідних текстів, необхідні зовнішні модулі, цілі збірки).
- 2) Цей файл передається утиліті CMake.
- 3) Результатом роботи CMake є файл, що містить інструкції складання програми для конкретної платформи (make-файл GNU make, файл проекту Visual Studio і т.д.). Суть ідеї полягає в тому, що опис процесу складання у файлі CMakeLists.txt абстраговано як від конкретних особливостей окремих систем (розташування файлів, можливості компіляторів), так і цілих платформ.

- 4) Читаючи загальний опис процесу складання з файлу CMakeLists.txt, програма CMake створює файл інструкцій збірки, що враховує специфіку конкретної системи.

Що ж забезпечує таку універсальність CMake ?

Це легко пояснюється тим, що в її основі лежить потужна мова сценаріїв, яка використовується як в файлах CMakeLists.txt, так і в спеціальних сценаріях - модулях. Пошук файлів і перевірка можливостей системи реалізовані саме в модулях. Щодо файлів модулів, то вони розташовані в піддиректорії Modules директорії CMake.

На даний момент CMake доступна як для користувачів Linux (встановлення і робота через командну стрічку) так і Windows (exe – файл інсталяції). Усі дистрибутиви є абсолютно безкоштовними і їх можна завантажити із офіційного сайту: <https://cmake.org>.

Для користувачів Windows перевага полягає у тому, що CMake має графічний інтерфейс, що робить роботу із ним набагато зручнішою. Відповідно і назва – CMake GUI.

Для створення проекту веб-додатку із використанням фреймворку Wt у Visual Studio, відповідно на платформі Windows використовуємо саме графічну версію.

У нашому випадку для генерування усіх, необхідних для генерування проекту файлів перш за все необхідно, щоб у програмі Wt Designer був вказаний вірний шлях до Wt-бібліотек у вкладці CMake (див. на рис. 1.3).

Пошук файлів і перевірка можливостей системи реалізовані саме в модулях. Щодо файлів модулів, то вони розташовані в піддиректорії Modules директорії CMake.

У нашому випадку для генерування усіх, необхідних для генерування проекту файлів перш за все необхідно, щоб у програмі Wt Designer був вказаний вірний шлях до Wt-бібліотек у вкладці CMake (див. на рис. 1.3).

					ТНТУ КПКН 16.027.27.000 ПЗ	Арк.
						23
Зм.	Арк.	докум.	Підп.	Дата		

Фінальним етапом є обрання каталогу із створеними раніше файлами, що містять інструкції, щодо генерування, а також вказати каталог, у котрий будуть збережені файли «конвертованого» проекту. Потім натискаємо кнопку Configure, обираємо відповідне середовище розробки та параметри щодо компіляторів і натискаємо Finish. Розпочнеться процес створення проекту, про перебіг якого буде відображено. У разі успіху – Configure Done.

На завершення клікаємо Generate, після успішної генерації отримуємо повноцінний проект, який побудується і скомпілюється у Microsoft Visual Studio.

На рис. 1.5 наведено базові налаштування CMake.

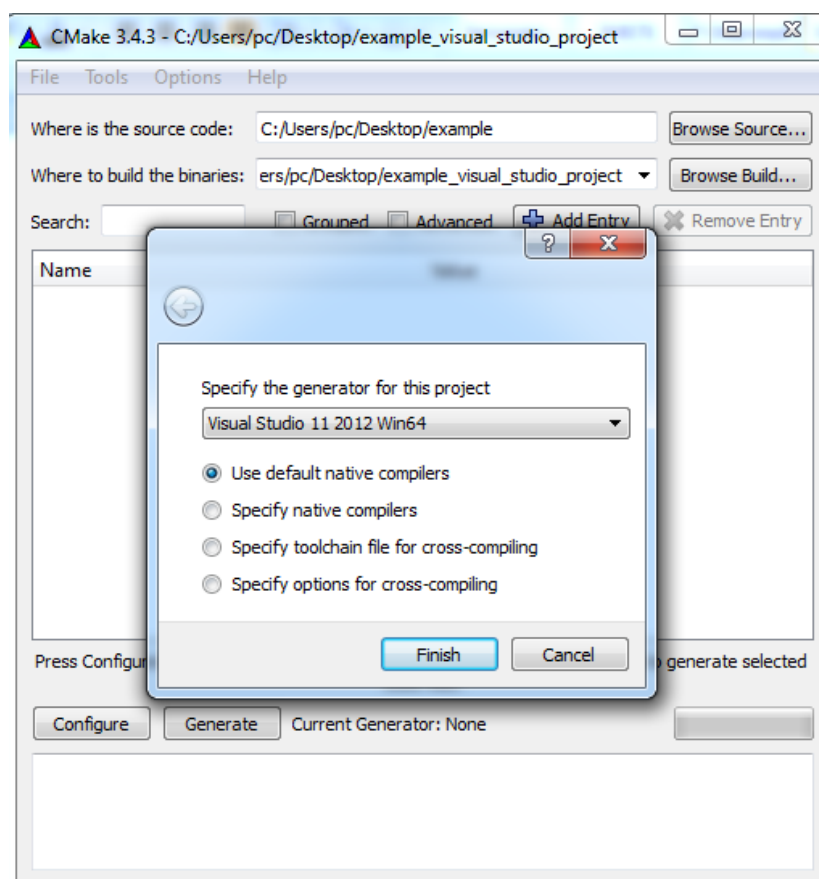


Рисунок 1.5 – Базові налаштування CMake

На рис. 1.6 наведено вигляд вікна утиліти-генератора, власне після процесу генерації.



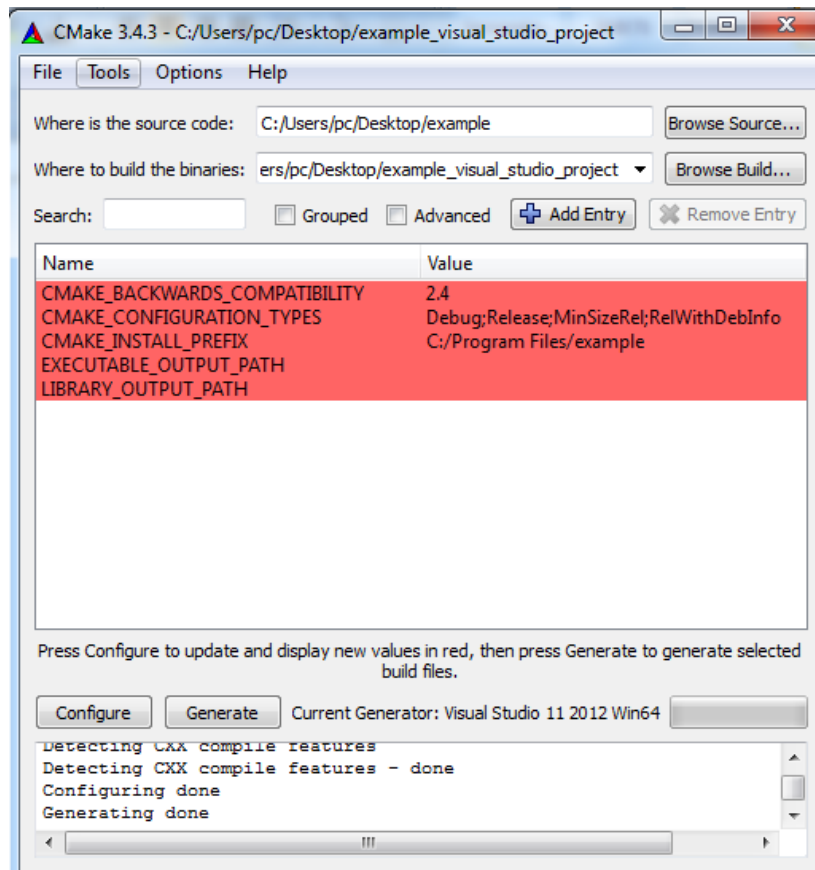


Рисунок 1.6 – Повідомлення про успішну конфігурацію та генерацію

Отже, CMake є надзвичайно корисною, універсальною та потужною програмою, яка дозволяє створювати програмні продукти для різних платформ, власне генеруючи проекти. Також вона дозволяє виконувати як in-place, так і out-of-place збірки. Тобто можна виконувати кілька збірок із одного дерева програмного коду, а також виконувати крос-платформну компіляцію. Крім того, можливість створювати дерево каталогів збірки за межами вихідного дерева є ключовою особливістю, це гарантує, що якщо побудований каталог видалений, вихідні файли залишаються без змін.

## РОЗДІЛ II

### ОБґРУНТУВАННЯ АЛГОРИТМУ І СТРУКТУРИ ПРОГРАМИ

Для можливості користувачем після оцінювання фільмів дізнатись із яким іншим користувачем його вподобання схожі застосовується метод інверсії.

Суть і алгоритм цього методу розглянемо детальніше.

Створений веб-додаток надає своїм користувачам можливість перегляду фільмів онлайн. Періодично система надає нові рекомендації користувачам — які фільми, що їх користувач ще не дивився, можливо будуть йому або їй цікаві, а також із яким саме користувачем його вподобання майже або повністю співпадають.

В основі методу інверсій лежить ідея, що користувачі, які подивились однакові фільми та також оцінили їх схожим чином, мають схожі смаки. Наприклад, нехай є два користувача: Оксана та Дмитро. Обидва вони переглянули наступні фільми (також вкажемо номери фільмів для подальшого посилання): “Зоряні війни” (1), “Гравітація” (2), “Пірати Карибського моря” (3), “Володар перснів” (4), “Матриця” (5).

Спочатку система просить користувачів оцінити ці фільми і розташувати їх у порядку вподобання., іншими словами — створити власний хіт-парад. Так, Оксана розташувала вказані фільми у порядку від найбільш до найменш вподобаного: “Пірати Карибського моря”, “Володар перснів”, “Матриця”, “Гравітація”, “Зоряні війни”. Іншими словами це можна описати так: фільм №1 в Оксани потрапив на позицію 5, фільм №2 - на позицію 4, фільм №3 - на позицію 1, фільм №4 - на позицію 2 та фільм №5 - на позицію 3.

Хіт-парад Дмитра: “Зоряні війни”, “Володар перснів”, “Гравітація”, “Матриця”, “Пірати Карибського моря” (або іншими словами: фільм №1 - на по

					ТНТУ КПКН 16.027.27.000 ПЗ								
Зм.	Арк	докум.	Підп.	Дата									
Розроб.		Палка О. В.			РОЗДІЛ II				Літ.	Арк.	Аркушів		
Перевір.		Бревус В. М.									26	39	
									ТНТУ, ФІС, СН-22				
Зав. каф.													

зицію 1, №2 - на позицію 3, №3 - на позицію 5, №4 - на позицію 2, №5 - на позицію 4).

На рис. 2.1 наведено порівняльну таблицю вподобань двох користувачів.

		Позиції фільмів у хіт-парадах	
	Фільм	Оксана	Дмитро
1	Зоряні війни	5	1
2	Гравітація	4	3
3	Пірати Карибського моря	1	5
4	Володар перснів	2	2
5	Матриця	3	4

Рисунок 2.1 – Оцінювання фільмів користувачами

Після цього система може надати кількісну оцінку, наскільки схожими є смаки двох користувачів. Для цього використовується алгоритм підрахунку інверсій поміж двома масивами.

*Нехай  $A[1..n]$  — масив з  $n$  чисел. Якщо  $i < j$  та  $A[i] > A[j]$ , то пара  $(i, j)$  — інверсія в  $A$ .*

Нам необхідно врахувати як далеко знаходяться один від одного два масиви (хіт-паради для Оксани та Дмитра). Щоб відчуті цю відстань, подивимось на фільм “Пірати Карибського моря”: в Оксани він на першому місці, а в Дмитра - на останньому. Тому тільки цей фільм окремо повинен вносити у загальну відстань між масивами значення  $5-1=4$ .

Таким чином, щоб звести задачу порівняння двох хіт-парадів до задачі підрахунку інверсій у нашому прикладі, побудуємо масив  $A$  наступним чином: елементом  $A[j]$  є число, яке відповідає позиції фільму в хіт-параді Дмитра, який в хіт-параді Оксани посідав місце  $j$ . Наприклад,  $j = 1$  у хіт-параді Оксани відповідає фільму “Пірати карибського моря”. Цей фільм в списку Дмитра стоїть на позиції 5, тому  $A[1] = 5$ . Отже, загалом отримуємо масив  $A = [5, 2, 4, 3, 1]$ , утворення якого наведено на рис. 2.2.

Оксана	1	2	3	4	5
Дмитро	5	2	4	3	1

← Новий масив A

Рисунок 2.2 – Формування «результуючого» масиву для пошуку інверсій

*Іншими словами, значення елементів  $x_{it}$ -параду Оксани стали індексами нового масиву  $A$ , тоді як значення елементів  $x_{it}$ -параду Дмитра - значеннями цього масиву  $A$ .*

Масив  $A = [5, 2, 4, 3, 1]$  має наступні інверсії (вказуються індекси елементів, а не їх значення – див. на рис. 2.2, де це є елементи першого рядка): (1,2), (1,3), (1,4), (1,5), (2,5), (3,4), (3,5), (4,5). Загалом 8 інверсій. І це число вказує наскільки сильно відрізняється список вподобань Оксани від списку вподобань Дмитра.

Сервіс перегляду фільмів онлайн має базу даних  $D$  вподобань користувачів. Ця база є матрицею. Рядки цієї матриці відповідають користувачам, а стовпці — фільмам. Її розмірність  $ixm$ , де  $i$  — це кількість користувачів,  $m$  — кількість фільмів. Кожний елемент матриці  $D[i, j]$  вказує на позицію фільму  $j$  в списку вподобань користувача  $i$ . Для спрощення припускаємо, що всі користувачі переглянули всі фільми.

Тепер, щоб визначити наскільки подібні смаки деякого користувача  $x$  до смаків інших користувачів, система попарно порівнює списки вподобань  $x$  та всіх інших користувачів  $i \neq x$ : за вказаним вище принципом підраховується кількість інверсій у масиві  $D[x]$  відносно масиву  $D[i]$ . Визначене число інверсій буде кількісною оцінкою наскільки смаки  $x$  є близькими до смаків кожного  $i$  — чим менше значення цього числа, тим більш подібними є смаки двох користувачів.

Структура програми полягає у тому, що вона складається із графічного інтерфейсу – вигляду веб-сторінки, доступного для користувачів і, власне, передбачає наявність приватних та захищених полів і методів у класах `Films` і `Users`, які «координуватимуть і керуватимуть» роботою додатку (дії при натисканні кнопки, запис оцінок у файл, зчитування із файлу та підрахунків інверсій, вибір мінімальної інверсії для найбільш знаходження кіноманів із подібними смаками).

## РОЗДІЛ III РОЗРОБКА ПРОГРАМИ

Веб-додаток містить такі класи, як: *Ui\_Films*, *Films* і *Users*. Діаграму класів разом із їх методами наведено на рис. 3.1.



Рисунок 3.1 – Діаграма класів

Клас *Ui\_Films* (на рис. 3.2) виконує розмітку веб-сторінки (фронтент), проте також реагує на дії користувачів за допомогою методів . У цьому класі представлені п'ять методів:

- Метод *void setupUi(Wt::WContainerWidget \*PageRoot)* – ініціалізує створення головної сторінки і її побудову із застосуванням контейнерів, віджетів, кнопок, підключених файлів стилів, фреймворку Bootstrap 3;
- Метод *void addAllStyleSheets()* – необхідний для підключення CSS-файлу розмітки сторінки;
- Метод *void addAllJavaScripts()* – необхідний для підключення JavaScripts-файлу розмітки сторінки;
- Метод *void addAllJavaScripts()* – необхідний для підключення JavaScripts-файлу розмітки сторінки;
- Метод *void connectAllSignals()* – визначає поведінку додатку при певних «сигналах» - введення тексту, натискання кнопок;

					ТНТУ КПКН 16.027.27.000 ПЗ		
Зм.	Арк	докум.	Підп.	Дата	<div style="display: flex; justify-content: space-between; align-items: center;"> <div>РОЗДІЛ III</div> <div>ТНТУ, ФІС, СН-22</div> </div>		
Розроб.		Палка О. В.					
Перевір.		Бреус В. М.					
Зав. каф.							
					Літ.	Арк.	Аркушів
						29	39

- Метод `void writeFile(char* user, char* film_ID, char* mark)` – записує у файл дії користувача при натисканні клавіш під час етапу оцінки фільму.

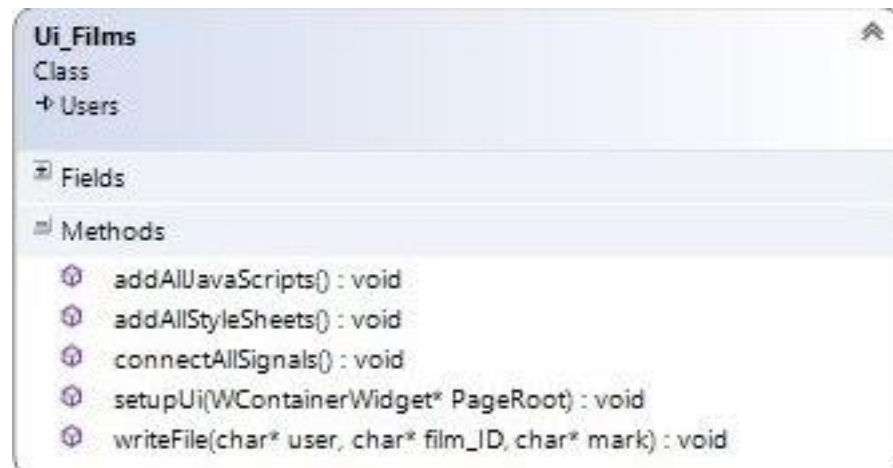


Рисунок 3.2 – Клас *Ui\_Films*

Важливим також є клас *Films* (на рис. 3.3) є нащадком класу *Wt::WApplication* і призначений для зберігання вказівника на об'єкт *ui* класу *Ui\_Films* у приватному полі, щоб підтримувалась така парадигма ООП як інкапсуляція – користувач не повинен мати доступу до усіх полів і методів, щоб не зіпсувати реалізацію. Конструктор даного класу видаляє файл із оцінками користувачів після завершення роботи програми, тобто вона не залишає після себе залишкових файлів.

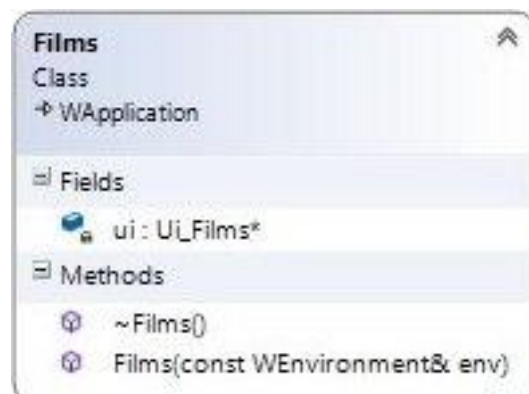


Рисунок 3.3 – Клас *Films*

Щодо класу *Users* (на рис. 3.4), то він забезпечує зчитування із файлу оцінок користувачів та за допомогою методу інверсії дозволяє дізнатись корис-

тувачів із схожими вподобаннями. Ці операції забезпечуються двома методами:

- Метод *string ReadFromFile\_and\_result (int userNumber)* – зчитує дані про оцінки із файлу, записує їх у 4 окремі масиви, адже додаток розрахований на 4 користувачів, викликає функцію пошуку інверсій між користувачами, знаходить мінімальне значення між інверсіями і вказує із яким користувачем вподобання найбільш співпадають;
- Метод *int inversion (int \*a,int \*b,int f)* – забезпечує знаходження числа інверсій між обраними користувачами. Код до даної функції наведено у додатку А.

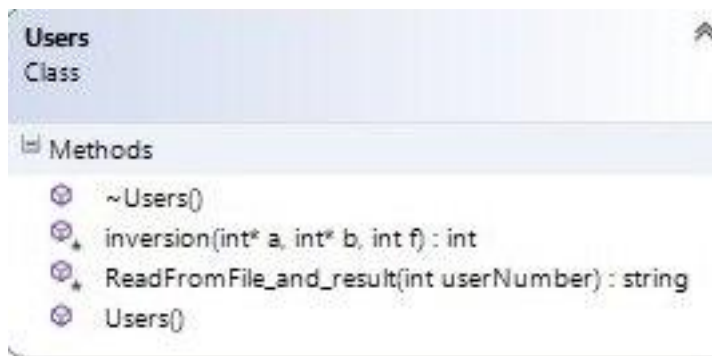


Рисунок 3.4 – Клас *Users*

Робота усіх цих класів із своїми методами і полями забезпечує запуск та роботу веб-додатку, написаного мовою програмування C++ із використанням фреймворків Wt і Bootstrap 3, програмних засобів CMake та WtDesigner.

## РОЗДІЛ IV

### ТЕСТУВАННЯ ПРОГРАМИ І РЕЗУЛЬТАТИ ЇЇ ВИКОНАННЯ

Запуск додатку і, відповідно, серверу відбувається у вигляді консольної програми (на рис. 4.1). Щодо вікна самої програми, то воно завантажується у будь-якому браузері, ввівши у адресну стрічку посилання **localhost:8080** (на рис. 4.2). Після надсилання запитів на **localhost:8080** у консольному додатку спостерігається процес надсилання пакетів, завантаження вмісту сторінок (на рис. 4.3).

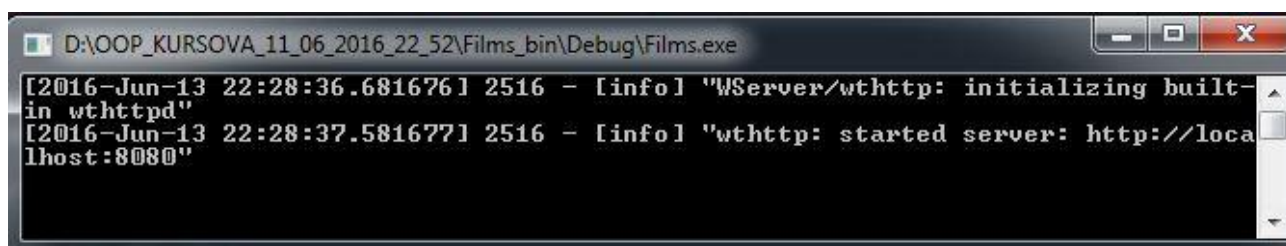


Рисунок 4.1 – Вигляд запущеного додатку у консолі (запуск серверу)

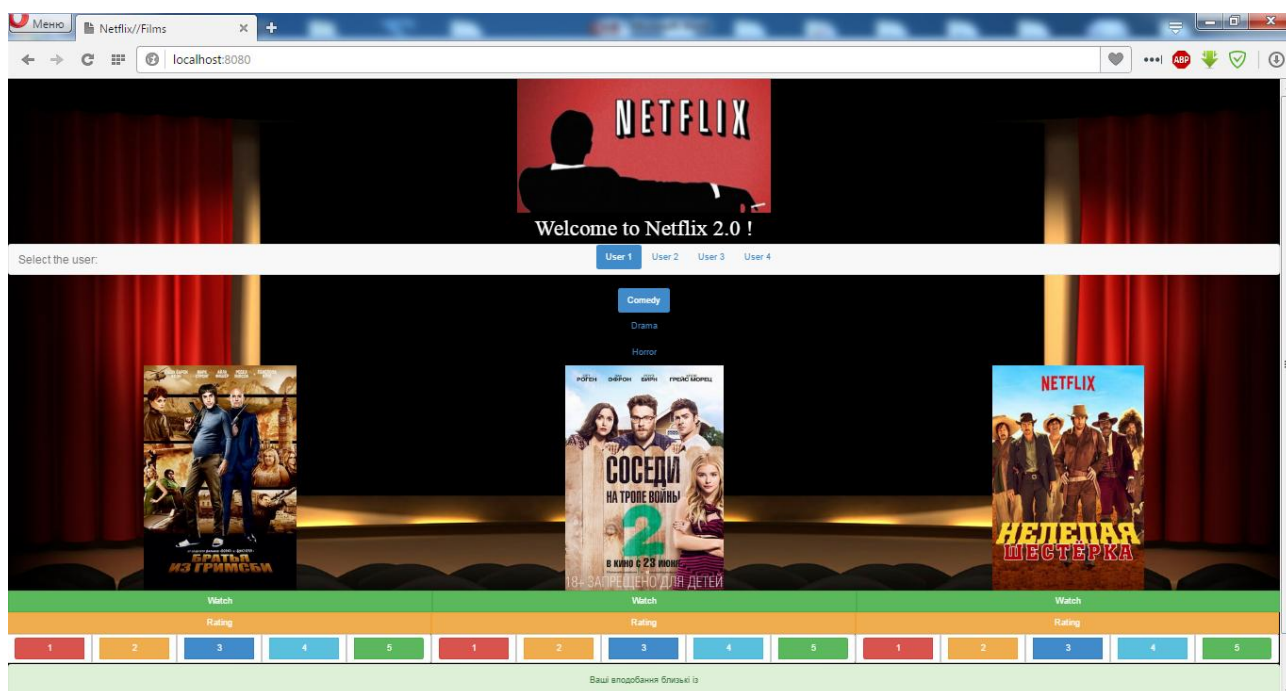


Рисунок 4.2 – Веб-сторінка, завантажена у браузері

					ТНТУ КПКН 16.027.27.000 ПЗ		
Зм.	Арк	докум.	Підп.	Дата			
Розроб.		Палка О. В.			РОЗДІЛ IV		
Перевір.		Бреус В. М.					
Зав. каф.							
						Літ.	Арк.
							32
						Аркушів	39
						ТНТУ, ФІС, СН-22	



```

D:\OOP_KURSOVA_11_06_2016_22_52\Films_bin\Debug\Films.exe
[2016-Jun-13 22:28:36.681676] 2516 - [info] "WServer/wthttp: initializing built-
in wthttpd"
[2016-Jun-13 22:28:37.581677] 2516 - [info] "wthttp: started server: http://loca
lhost:8080"
[2016-Jun-13 22:55:15.087022] 2516 - [info] "Wt: session created (#sessions = 1)
"
[2016-Jun-13 22:55:15.551049] 2516 [/ 5DTNGhWPYIxaY14] [info] "WEnvironment: Us
erAgent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/51.0.2704.63 Safari/537.36 OPR/38.0.2220.29"
::1 - - [2016-Jun-13 22:55:17.831179] "GET / HTTP/1.1" 200 2921
[2016-Jun-13 22:55:18.001189] 2516 - [info] "WebRequest: took 3123.18ms"
[2016-Jun-13 22:55:22.171427] 2516 [/ 5DTNGhWPYIxaY14] [warning] "WContainerWid
get: addWidget(): reparenting widget"
::1 - - [2016-Jun-13 22:55:22.852466] "GET /?wtd=5DTNGhWPYIxaY14&request=style&
page=1 HTTP/1.1" 200 103
[2016-Jun-13 22:55:22.864467] 2516 - [info] "WebRequest: took 4698.27ms"
::1 - - [2016-Jun-13 22:55:23.308492] "GET /resources/main.css HTTP/1.1" 304 0
::1 - - [2016-Jun-13 22:55:23.344494] "GET /resources/webkit-transitions.css HTT
P/1.1" 304 0
::1 - - [2016-Jun-13 22:55:23.346495] "GET /resources/themes/bootstrap/3/bootstr
ap.css HTTP/1.1" 304 0
::1 - - [2016-Jun-13 22:55:23.379496] "GET /resources/themes/bootstrap/3/wt.css
HTTP/1.1" 304 0
::1 - - [2016-Jun-13 22:55:24.218544] "GET /?wtd=5DTNGhWPYIxaY14&sid=-177180145
&webGL=true&scrW=1366&scrH=768&tz=180&htmlHistory=true&deployPath=%2F&request=sc
ript&rand=947683266 HTTP/1.1" 200 48905
[2016-Jun-13 22:55:24.244546] 2516 - [info] "WebRequest: took 5008.29ms"
::1 - - [2016-Jun-13 22:55:24.853581] "POST /?wtd=5DTNGhWPYIxaY14 HTTP/1.1" 200
496
[2016-Jun-13 22:55:24.860581] 2516 - [info] "WebRequest: took 54.003ms"

```

Рисунок 4.3 – Детальний процес обробки даних (GET і POST запити)

Для кожного із 4 користувачів є три категорії фільмів, серед яких: «Комедії», «Драми» та «Жахи». На рис. 4.4 наведено категорію «Драми» для User 2. На рис. 4.5 наведено категорію «Жахи» для User 1.

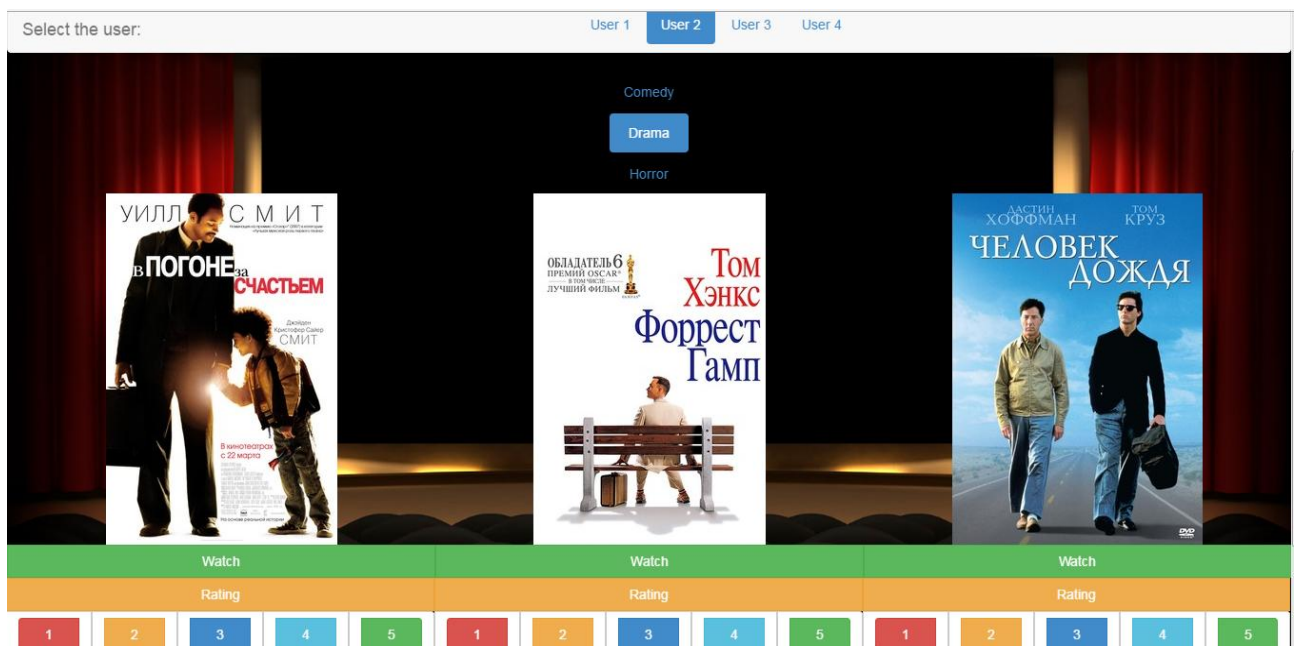


Рисунок 4.4 – Фільми категорії драми для User 2

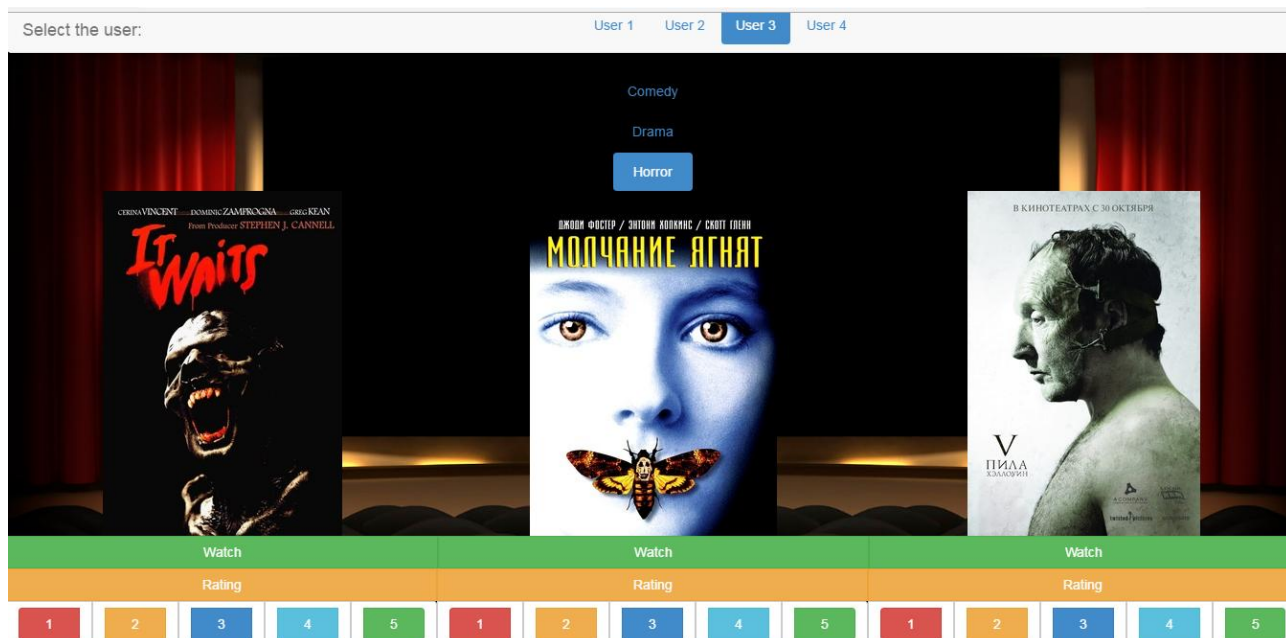


Рисунок 4.5 – Фільми категорії жахи для User 3

Також до кожного фільму є 7 кнопок, серед яких виділяються кнопки «Watch» і «Rating», адже вони переходять за посиланням на перегляд фільму онлайн і, відповідно на веб-ресурс, який спеціалізується на оцінках фільмів. На рис. 4.6 наведено перехід для перегляду фільму «Людина дощу» з категорії «Драми» для User 4, а на рис. 4.7 – перегляд рейтингу фільму «Мовчання ягнят» із категорії «Жахи».



Рисунок 4.6 – Перегляд фільму «Людина дощу» з категорії «Драми»

Зм.	Арк.	№ докум.	Підп.	Дата

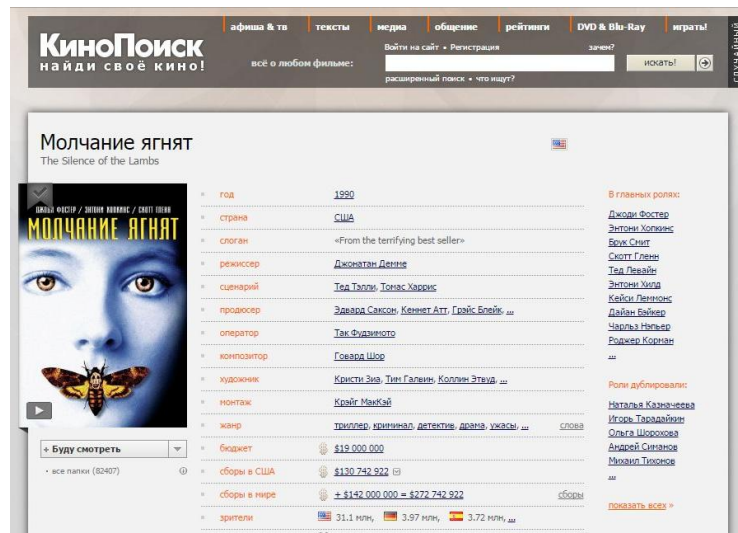


Рисунок 4.7 – Перегляд рейтингу фільму «Мовчання ягнят» із категорії «Жахи»

Коло кожного із фільмів після кнопки «Rating» розміщено п'ять пронумерованих кнопок, які після натискання змінюють вигляд і колір, стаючи неактивними і сигналізуючи про вже здійснено користувачем оцінку даного фільму. На рис. 4.8 продемонстровано процес клікання по кнопках, тобто оцінювання фільмів категорії «Комедії» для User 1. Важливою характерною ознакою є те, що усі дії користувача фіксуються і записуються у текстовий файл (на рис. 4.9). Даний документ слугує своєрідною базою даних, що вміщує у кожному рядку три числа: перше із них – номер користувача, друге – номер фільму і третє – оцінка даного фільму.

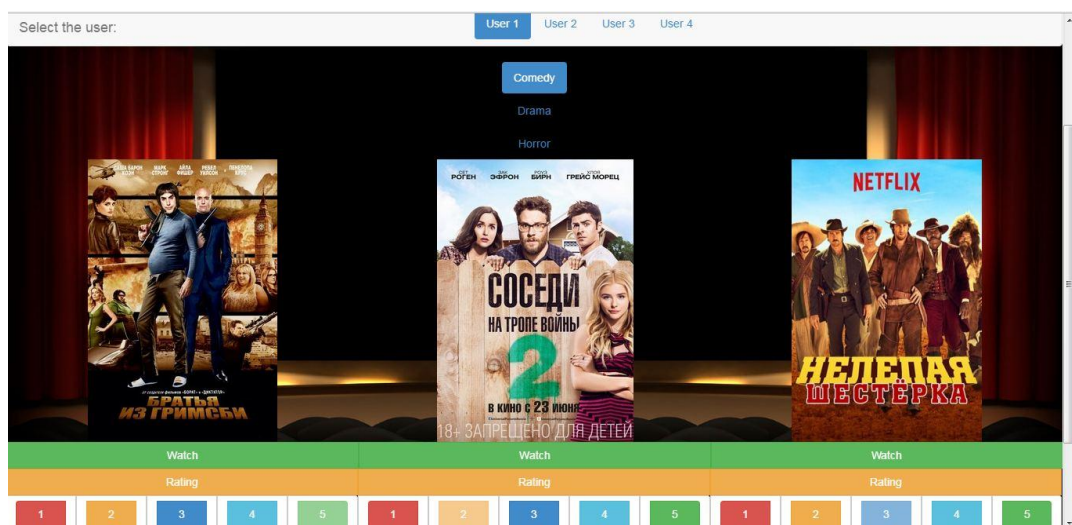


Рисунок 4.8 – Користувачьке оцінювання фільмів

Зм.	Арк.	№ докум.	Підп.	Дата



marks.txt				
1	1	1	5	
2	1	2	4	
3	1	3	4	
4	1	4	4	
5	1	5	5	
6	1	6	4	
7	1	7	4	
8	1	8	5	
9	1	9	3	
10	2	1	4	
11	2	2	3	
12	2	3	3	
13	2	4	5	
14	2	5	3	
15	2	6	3	
16	2	7	4	
17	2	8	5	
18	2	9	5	

Рисунок 4.9 – Оцінки у записаному файлі

На рис. 4.10 наведено вивід User-а із схожими вподобаннями для користувача №2.

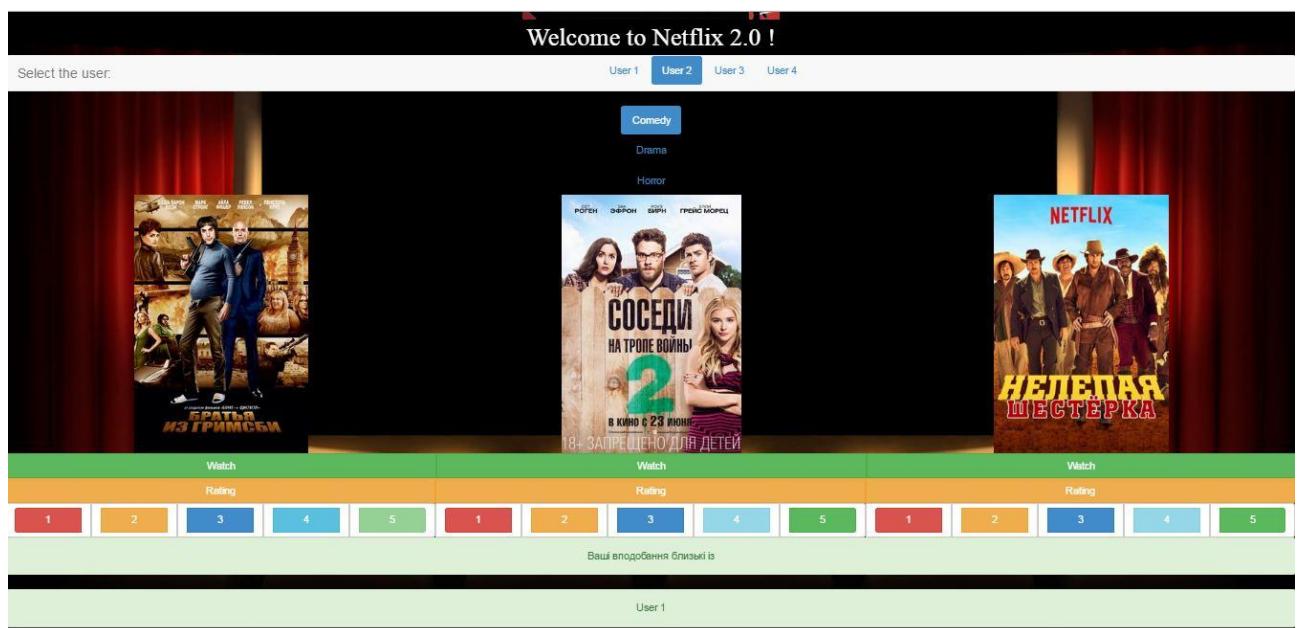


Рисунок 4.10 – Схожі вподобання, знайдені за допомогою інверсії

Отже, створений web-додаток «Netflix 2.0» вдало проходить етап компіляції, внаслідок чого сервер можливо запустити і отримати доступ до веб-сторінки, роботу із якою описано вище.

## ВИСНОВКИ

У результаті виконання роботи було створено web-додаток Netflix 2.0, призначений для оцінки та перегляну відеоматеріалів, а також для рекомендації користувачу фільмів, залежно від вподобань інших користувачів за допомогою методу інверсії.

Він розроблявся з використанням фреймворку Wt, Bootstrap 3, набору бібліотек Boost, програм WtDesigner і CMake у середовищі Microsoft Visual Studio 2012 мовою програмування C++.

Дана програма надає користувачеві можливість легко і зручно переглядати і оцінювати фільми. Завдяки інтуїтивно зрозумілому інтерфейсу будь-який користувач не зіткнеться з жодними проблемами при її використанні.

					ТНТУ КПКН 16.027.27.000 ПЗ			
Зм.	Арк	докум.	Підп.	Дата				
Розроб.		Палка О. В.			ВИСНОВКИ	Літ.	Арк.	Аркушів
Перевір.		Бревус В. М.					37	39
						ТНТУ, ФІС, СН-22		
Зав. каф.								

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Орлов С. А. Технологии разработки программного обеспечения: Учебник. - СПб.: Питер, 2002. – 464 с. ISBN: 5-94723-145-X(рус.).
2. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд. / Стивен Прата : Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2012. – 1248 с. ISBN 978-5-8459-1778-2 (рус.).
3. Страуструп Б. Язык программирования C++: Специальное издание. / Бьерн Страуструп. Пер. с англ. – М.: Издательство Бином, 2011. – 1136 с. ISBN 978-5-7989-0425-9 (рус.).
4. Дейтел Х.М. Как программировать на C++: 5-е издание. / Х.М. Дейтел, П. Дж. Дейтел : Пер. с англ. – М.: ООО «Бином-Пресс», 2008. – 1456 с. ISBN 978-5-9518-0224-8 (рус.).
5. Wt: anintroduction // Wt – a C++ Web Toolkit [Электронный ресурс]. – Режим доступа: <https://www.webtoolkit.eu/wt>. – Дата доступа: 13.04.2016.
6. Welcome to Boost.org! // Boost C++ LIBRARIES [Электронный ресурс]. – Режим доступа <http://www.boost.org>. – Дата доступа: 18.04.2016.
7. Bootstrap // Bootstrap [Электронный ресурс]. –Режим доступа: <http://getbootstrap.com>. – Дата доступа: 25.05.2016.

					ТНТУ КПКН 16.027.27.000 ПЗ			
Зм.	Арк	докум.	Підп.	Дата	<div style="text-align: center;"> <b>ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> </div>			
Розроб.		Палка О. В.						
Перевір.		Бревус В. М.						
Зав. каф.					<div style="text-align: center;"> ТНТУ, ФІС, СН-22 </div>			
					<div style="text-align: center;"> Літ.      Арк.      Аркушів </div>			
					<div style="text-align: center;"> 38      39 </div>			

Лістинг А1 – Метод *int inversion (int \*a,int \*b,int f)*

```

int Users::inversion (int *a,int *b,int f)
{
    int index;
    for(int i=0;i<f-1;i++)
    {
        index=i;
        for(int j=i+1;j<f;j++)
        {
            if(a[j]<a[index])
                index=j;
        }
        int buf=a[index];
        a[index]=a[i];
        a[i]=buf;
        buf=b[index];
        b[index]=b[i];
        b[i]=buf;
    }

    int rez=0;
    for(int i=0;i<f;i++)
        for(int j=i+1;j<f;j++)
            if(b[i]>b[j])rez++;
    return rez;
}

```

					ТНТУ КПКН 16.027.27.000 ПЗ		
Зм.	Арк	докум.	Підп.	Дата	ДОДАТОК А		
Розроб.		Палка О. В.					
Перевір.		Бреус В. М.					
Зав. каф.							
					Літ.	Арк.	Аркушів
						39	39
					ТНТУ, ФІС, СН-22		