





Dysk sieciowy

Przechowywanie danych w sieci jest wykorzystywane nie tylko w firmach, ale także przez setki milionów indywidualnych użytkowników nie związanych z branżą IT. Niektórym użytkownikom wystarcza relatywnie niewielka ilość danych 5..50GB, którą można przechowywać w sieci bezpłatnie. Jednak co roku coraz większa liczba internautów wybiera odpłatne usługi. Poniżej przedstawiono miesięczny koszt tej usługi w zależności od pojemności (2022-11-01).

Dyski sieciowe mogą być osobno płatną usługą lub częścią szerszej palety usług, za których opłatę wnosi się łącznie.

Niektóre popularne usługi sieciowe zmieniają się: <https://support.google.com/googleone/answer/9004014?hl=en>

Storage				
5 GB				
50 GB			4zł	
100 GB	\$2	9zł		\$2
200 GB		14zł	12zł	
1 TB	\$7			\$7
2 TB		47zł	40zł	\$12
6 TB	\$10			

Technologia dysków sieciowych wiąże się z technologiami chmurowymi opisanymi m.in.

https://pl.wikipedia.org/wiki/Chmura_obliczeniowa

https://pl.wikipedia.org/wiki/Chmura_publiczna

https://pl.wikipedia.org/wiki/Microsoft_Azure

https://en.wikipedia.org/wiki/Google_Cloud_Platform

<https://pl.wikipedia.org/wiki/iCloud>

https://pl.wikipedia.org/wiki/Amazon_Web_Services

Technologie wybranych dysków sieciowych

<https://pl.wikipedia.org/wiki/OneDrive>

https://pl.wikipedia.org/wiki/Dysk_Google i https://en.wikipedia.org/wiki/Google_One

<https://pl.wikipedia.org/wiki/iCloud>

https://en.wikipedia.org/wiki/Amazon_Drive

Porównanie wybranych dysków sieciowych

https://en.wikipedia.org/wiki/Comparison_of_file_hosting_services

System kontroli wersji Git

To ćwiczenie może okazać się przydatne dla osób realizujących w ramach pracy inżynierskiej aplikację, której rozwój, testowanie i optymalizację trzeba poprawnie udokumentować oraz profesjonalnie przebadac tę aplikację pod różnymi kątami, aby móc ją następnie zoptymalizować, zgodnie ze współczesnymi technologiami i trendami.

Projektowanie, programowanie, testowanie oraz dokumentowanie aplikacji sieciowej to złożony proces, nawet jeśli jest wykonywane tylko przez jedną osobę, jak to ma zwykle miejsce w przypadku pracy inżynierskiej. Stopień trudności tego zadania rośnie wykładniczo z każdym kolejnym programistą, który dochodzi do zespołu realizującego aplikację. Dlatego przy tworzeniu aplikacji od wielu lat wykorzystywane są systemy kontroli wersji (version/revision control system), czyli programy ułatwiające śledzenie zmian wprowadzanych w kodzie źródłowym i łączenie zmian realizowanych przez wiele osób na różnych etapach realizacji danego projektu.

https://pl.wikipedia.org/wiki/System_kontroli_wersji

Spośród wielu systemów kontroli wersji wyróżnia się projekt Git, stworzony przez Linusa Torvaldsa (twórcę Linuxa)

[https://pl.wikipedia.org/wiki/Git_\(oprogramowanie\)](https://pl.wikipedia.org/wiki/Git_(oprogramowanie)). Git jest rozproszonym systemem kontroli wersji.

To rozproszenie polega na tym, że na każdym komputerze pracującym w projekcie przechowywany jest cały projekt

i jego zmiany – tzw. repozytorium lokalne. System ten wyróżnia się większą niezawodnością, ponieważ nie ma głównego repozytorium przechowującego cały projekt, którego awaria mogłaby spowodować jego utratę. Git pełni rolę repozytorium zewnętrznego (remote), które pośredniczy między wszystkimi połączonymi maszynami i przechowuje cały projekt. Wyłączenie tego zewnętrznego repozytorium nie spowoduje utraty zapisanych zmian. Popularność Gita wynika m.in. z jego relatywnie dużej szybkości i niezawodności w porównaniu z konkurencyjnymi projektami, a także prowadzenia z jego pomocą bardzo ważnych lub/i dużych projektów m.in. jądra Linuxa, ponadto Git jest systemem darmowym. Instalacja Gita na lokalnym komputerze/serwerze jest opisana na wielu portalach podpowiadających, jak go skonfigurować krok-po-kroku np.

<https://git-scm.com/book/pl/v1/Pierwsze-kroki-Wprowadzenie-do-kontroli-wersji>

Więcej informacji o projekcie Git znajduje się m.in. na stronie:

<https://pl.wikibooks.org/wiki/Git>

<https://pl.wikibooks.org/wiki/Git/Podstawy>

Aby wykorzystywać projekt Git nie trzeba instalować oprogramowania na własnym komputerze/serwerze, ponieważ istnieje wiele serwisów hostingowych umożliwiających bezpłatne lub płatne utrzymywanie repozytoriów Gita np. <https://github.com/> i wiele innych, których cechy można przejrzeć i porównać korzystając m.in. z poniższych linków:

<https://git.wiki.kernel.org/index.php/GitHosting>

https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities

<https://www.git-tower.com/blog/git-hosting-services-compared/>

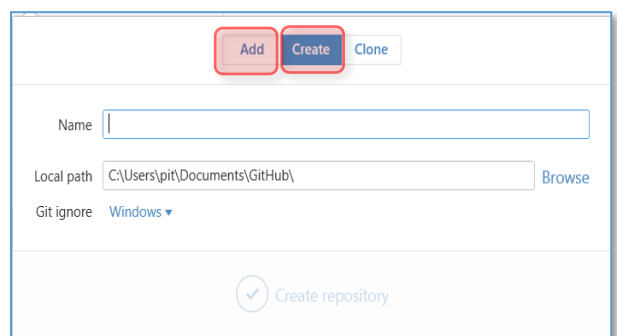
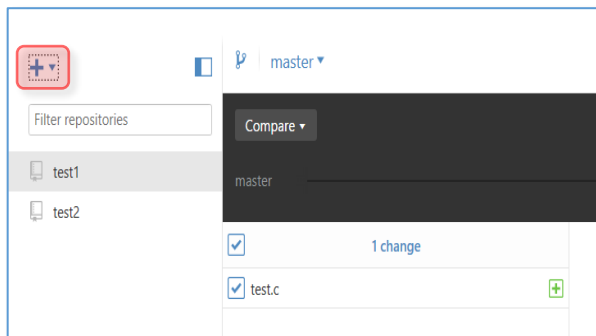
Korzystając z wymienionych aplikacji można utrzymywać kod rozwijany prywatnie lub udostępniany publicznie.

Prywatne repozytoria kodu służą firmom rozwijającym aplikacje np. wspólnie ze współpracującymi programistami z zewnątrz, natomiast dzięki publicznym repozytoriom wielu pracodawców poszukujących programistów do realizacji konkretnych projektów znajduje odpowiednich kandydatów lub weryfikuje możliwości potencjalnych pracowników, prosząc ich o podanie konta na Githubie, aby się przyjrzeć próbkom ich kunsztu programistycznego. Portal github.com jest zatem wykorzystywany zarówno przez wielkich developerów, jak i m.in. tzw. wolnych strzelców.

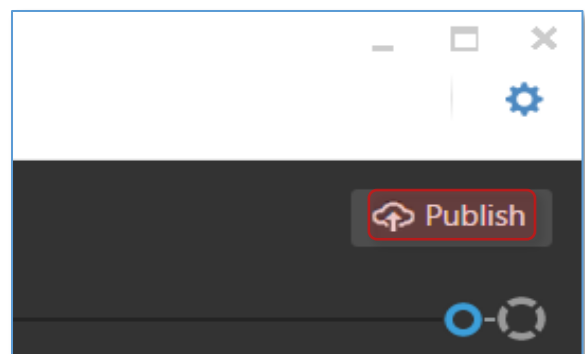
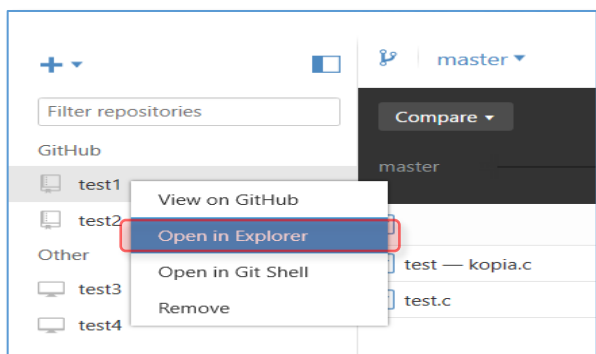
UWAGA: Poniższa instrukcja zakłada, że nie korzystałeś dotychczas z githuba. Jeśli Twoje narzędzia programistyczne są już skonfigurowane do współpracy z githubem – przejdź do właściwej części realizacji zadania.

1. Wejdź na portal <https://github.com/>
2. Utwórz swój prywatny projekt <https://github.com/personal> (jeśli posiadasz swoje konto na Githubie pomiń ten punkt i przejdź od razu do realizacji aplikacji, przy czym wykorzystuj Githuba przy jej tworzeniu).
3. Utwórz dla siebie bezpłatne konto, umożliwiające Ci dzielenie się swoim kodem ze światem lub przynajmniej potencjalnymi pracodawcami <https://github.com/join>
 - a. Utwórz nazwę swojego konta (Create your personal account). Zastanów się chwilę nad sensowną nazwą, którą można byłoby wykorzystać w przyszłości, w celu przesłania jej pracodawcy w swoim cv.
 - b. Podaj swój adres **email** (podaj istniejący email, trzeba go potem wykorzystać do potwierdzenia rejestracji).
 - c. Stwórz **hasło** dostępu do portalu (istnieją wymagania co do złożoności hasła, zapisz gdzieś to hasło).
 - d. Kliknij przycisk **Create an account**
 - e. Wybierz opcję **Unlimited public repositories for free**.
 - f. Naciśnij klawisz **Continue**
 - g. Wprowadź informacje o sobie: stopień zaawansowania w programowaniu, plany wykorzystania GitHuba, itd. lub pomiń ten krok.
 - h. Kliknij klawisz **Start a project**
 - i. Wejdź na swój serwer pocztowy, odbierz mail z Githuba i kliknij zawarty w nim link, aby potwierdzić utworzenie konta.
 - j. Utwórz nazwę repozytorium np. *MyCloud*
 - k. Utwórz opis repozytorium np. *Zadanie laboratoryjne MyCloud*
 - l. Wybierz opcję **Public** dla swojego repozytorium, aby uniknąć wnoszenia opłat za prywatne repozytorium.
 - m. Naciśnij klawisz **Create repository**.
4. Ze strony <https://desktop.github.com/> pobierz aplikację, za pomocą której będziesz synchronizować swoje lokalne wersje projektów z wersjami przechowywanymi na Githubie.
 - a. W zależności od posiadanego systemu operacyjnego - kliknij odpowiedni link **Download GitHub Desktop**
 - b. W przypadku systemu Windows zapisz na pulpicie aplikację **GitHubSetup.exe**, za pomocą której można ściągnąć właściwą aplikację GitHub Desktop.
 - c. Aplikacja zainstaluje się w katalogu **Napęd:\Users\Nazwa_Użytkownika\AppData\Local\GitHub**
 - d. Uruchom tę aplikację i zaloguj się do swojego konta na Githubie podając: **mail** oraz **hasło**.
 - e. Od tej chwili możesz używać synchronizacji katalogów, w których trzymasz źródła aplikacji na swoim komputerze oraz katalogi na Githubie.






- f. Kliknij na przycisk '+' znajdujący się w lewej górnej części aplikacji



- g. Utwórz nowe repozytoria o nazwach test1, a potem test2 i test3, możesz to zrobić przez wydanie polecenia dodaj (Create) lub wskaż (Add) nowy katalog git. Jeśli wskazany katalog nie zawiera systemowych plików Gita (.gitignore, .gitattributes, .git – podkatalog z całą zawartością), a tak się stanie, gdy wskażemy katalog właśnie utworzony np. na pulpicie – wówczas aplikacja poprosi o zezwolenie na ich utworzenie.
- h. Kliknij na jednym z utworzonych repozytoriów test1 i wybierz opcję **Open in Explorer**. Explorer plików otworzy katalog skojarzony z danym repozytorium. Utwórz w tym katalogu przykładowy plik test.c, zwróć uwagę, że aplikacja wyświetliła informację o zmianie, która nastąpiła we wskazanym repozytorium, jeśli skopiujesz plik test.c, wówczas kolejna zmiana w repozytorium zostanie wyświetlona.



- i. Dopóty dopóki dane repozytorium nie zostanie opublikowane, będzie ono przechowywane jedynie na lokalnym komputerze. Aby to zmienić należy kliknąć na ikonę **Publish**, znajdującą się w prawym górnym rogu aplikacji. Dopiero wówczas odblokowuje się menu kontekstowe danego repozytorium pozwalając otworzyć je na Githubie **View on GitHub**. Wykonaj szereg eksperymentów z tworzeniem repozytoriów, ich publikowaniem, synchronizacją oraz usuwaniem. Zwróć uwagę na informacje wyświetlane lokalnie przez aplikację **GitHub Desktop** oraz na informacje wyświetlane przez portal github.com.
- j. Przecwiczenie tych opcji jest bardzo ważne, aby zrozumieć wszystkie mechanizmy działania Githuba i potem swobodnie z niego korzystać, bez obawy, że zgubimy gdzieś przez przypadek jaką ważną wersję tworzonej aplikacji, do której będzie trzeba wrócić.
5. Wykorzystując program kontroli wersji Github stwórz aplikację do przechowywania prywatnych plików na serwerze hostingowym, wzorowaną na aplikacjach firm Google, Microsoft, Apple oraz Amazon.
6. Funkcjonalności tworzonego dysku sieciowego
(**Uwaga: niektóre wymienione poniżej funkcje były całkowicie lub częściowo realizowane w poprzednich ćwiczeniach**)
- Rejestrowanie się i logowanie się użytkowników,
 - Uwaga:** Wykorzystaj skrypty do logowania/rejestracji/sesji z poprzednich zadań.
 - Hasła użytkowników mają być trzymane w tabeli **users**.
 - Rejestrowanie daty i godziny pomyślnych oraz błędnych prób logowań na konta
 - Uwaga:** Wykorzystaj skrypty rejestrowania logowań do portalu z poprzednich zadań,
 - Odpowiednio zmodyfikuj te skrypty wg wskazówek zawartych w tej instrukcji.
 - Informacje o logowaniu mają być trzymane w tabeli o nazwie **goscieportal**.
 - Wprowadź limit ilości błędnych logowań występujących po sobie – po 3 błędnych logowaniach dane konto jest blokowane na **1 minutę**, ma to służyć utrudnieniu łamania hasła metodą Brute-force.

- v. Trzykrotne błędne logowanie ma być dodatkowo zarejestrowane w osobnej tabeli o nazwie [break_ins](#), przechowującej datę/godzinę tego zdarzenia i IP komputera, z którego nastąpiła nieudana próba logowania. Zbieranie tego typu danych umożliwia działania proaktywne m.in. tworzenie *białych list* i *czarnych list*, a w konsekwencji np. blokowanie dostępu do portalu z określonych adresów IP, domen itd. Zawartość tej tabeli będzie udostępniona administratorowi tej aplikacji.
 - vi. Po trzykrotnym błędnym logowaniu nastąpi blokada logowania na 1 minutę, a gdy potem nastąpi poprawne logowanie – wówczas logującemu się użytkownikowi ma się wyświetlić **na czerwono** komunikat ostrzegawczy o dacie i godzinie ostatniego błędnego logowania oraz IP komputera, z którego te próby były wykonane. Takie rozwiązanie stosowane jest w niektórych portalach internetowych umożliwiających m.in. dostęp do kont i transakcji bankowych.
- c. Po pomyślnym zarejestrowaniu się użytkownika tworzony jest dla niego na serwerze [katalog macierzysty](#) o nazwie takiej, jak login użytkownika.
- i. **Uwaga:** Wykorzystaj skrypty dotyczące tworzenia [katalogu macierzystego](#) z poprzedniego zadania.
 - ii. W skład loginu użytkownika mogą wchodzić jedynie znaki (m.in. litery i cyfry) dopuszczone do tworzenia z nich nazw katalogów.
 - iii. Wszystkie operacje dyskowe zalogowanego użytkownika odbywają się wewnątrz tego [katalogu macierzystego](#), stąd po pomyślnym zalogowaniu warto utworzyć zmienną przechowującą tę ścieżkę, bo będzie ona potrzebna do różnych operacji.
- d. Wyświetlanie listy plików i podkatalogów [katalogu macierzystego](#),
- i. ta operacja ma się odbywać automatycznie po zalogowaniu się użytkownika do jego dysku sieciowego,
 - ii. po pierwszym zalogowaniu danego użytkownika lista plików i podkatalogów jego [katalogu macierzystego](#) będzie pusta.
- e. Wyświetlanie listy plików podkatalogu [katalogu macierzystego](#), do którego wszedł użytkownik
- i. ta operacja odbywa się automatycznie po wejściu użytkownika do danego podkatalogu [katalogu macierzystego](#),
 - ii. wejście do podkatalogu jest zrealizowane przez kliknięcie jego nazwy, która jest linkiem do skryptu realizującego tę funkcję.
- f. Tworzenie podkatalogów [katalogu macierzystego](#)
- i. ta operacja jest realizowana po kliknięciu ikony , uruchamiającym skrypt umożliwiający podanie nazwy nowego podkatalogu i tworzącym go na serwerze wewnątrz [katalogu macierzystego](#) <https://www.php.net/manual/en/function.mkdir.php>,
 - ii. przyjmij zasadę, że w [katalogu macierzystym](#) można utworzyć dowolną ilość podkatalogów, ale wewnątrz tych podkatalogów nie tworzymy dalszych podkatalogów, dlatego ikona  ma być wyświetlana jedynie wówczas, gdy użytkownik jest w swoim [katalogu macierzystym](#), a nie w którymś z jego podkatalogów.
- g. Usuwanie plików lub podkatalogów [katalogu macierzystego](#)
- i. ta operacja jest realizowana po kliknięciu ikony , wyświetlanej przy nazwie każdego pliku i podkatalogu,
 - ii. kliknięcie tej ikony uruchamia skrypt usuwający dany plik lub podkatalog <https://www.php.net/manual/en/function.unlink.php>, <https://www.php.net/manual/en/function.rmdir.php>,
 - iii. przy usuwaniu podkatalogu zawierającego wewnątrz jakieś pliki, należy je najpierw usunąć.
- h. Wchodzenie do podkatalogów [katalogu macierzystego](#) i wychodzenie z nich
- i. wejście do podkatalogu następuje przez kliknięcie jego nazwy, która jest linkiem do skryptu realizującego tę funkcję,
 - ii. powrót z podkatalogu do [katalogu macierzystego](#) następuje przez kliknięcie ikony level up ,
 - iii. ikona ta jest wyświetlana, gdy użytkownik znajduje się w podkatalogu swojego [katalogu macierzystego](#).
- i. Przesyłanie (upload) plików do [katalogu macierzystego](#) lub jego podkatalogów lub pobieranie (download) z nich plików
- Uwaga:** Wykorzystaj odpowiednie skrypty z poprzednich zadań, m.in. [select.php](#), [upload.php](#) oraz skrypty do odtwarzania w portalu plików multimedialnych.
- i. przesyłanie pliku (upload) składa się z następujących działań
 - wejścia do danego podkatalogu lub pozostania w [katalogu macierzystym](#), jeśli w nim ma być realizowana operacja,
 - kliknięcia ikony file upload , która uruchamia skrypt umożliwiający wybór pliku z lokalnego komputera, w celu przesłania go na serwer.

- ii. pobieranie pliku (download) z danego katalogu składa się z następujących działań
 - wejścia do danego podkatalogu lub pozostania w [katalogu macierzystym](#), jeśli w nim ma być realizowana operacja,
 - kliknięcia nazwy pliku, który ma być pobrany, nazwa pliku powinna być hiperłączem do pliku,
 - ściągnięcia z serwera wybranego pliku na lokalny komputer, co w praktyce jest realizowane automatycznie przez przeglądarkę, o ile wyświetlona nazwa/ikona pliku wskazuje prawidłowe hiperłącze do tego pliku,
- iii. kod programu powinien być tak stworzony, aby pliki wyświetlały się / odtwarzały się poprawnie
 - pliki multimedialne (graficzne, filmy i muzyka) wyświetlają się w postaci nazw plików, obok których ma być wyświetlana miniatura pliku graficznego lub odtwarzacz mp3 lub odtwarzacz mp4,
 - pozostałe pliki wyświetlają się jedynie w formie nazwy pliku z rozszerzeniem,
 - pliki graficzne powinny wyświetlać się w postaci miniatury, po kliknięciu miniatury obrazek powinien wyświetlić się w pełnym rozmiarze z poziomu aplikacji, bez konieczności ściągnięcia go na lokalny komputer,
 - filmy nie powinny odtwarzać się automatycznie, powinna być możliwość odtworzenia ich z poziomu aplikacji, bez konieczności ściągnięcia ich na lokalny komputer,
 - pliki dźwiękowe nie powinny odtwarzać się automatycznie, powinna być możliwość odtworzenia ich z poziomu aplikacji, bez konieczności ściągnięcia ich na lokalny komputer,
 - pliki, których formaty nie są wspierane przez przeglądarkę – powinny być ściągane na lokalny dysk po kliknięciu ich nazw w aplikacji.

7. Wykonaj szereg testów weryfikujących poprawność działania programu.

- a. Zarejestruj nowego użytkownika [test1](#),
 - i. zaloguj się na jego konto i prześlij do jego [katalogu macierzystego](#) kilka plików mp3, mp4, jpg, png, pdf i docx,
 - ii. utwórz kilka podkatalogów [katalogu macierzystego](#) i umieść w nich kilka różnych plików,
 - iii. sprawdź, czy możesz poprawnie poruszać się między [katalogiem macierzystym](#), a jego podkatalogami.
- b. Analogicznie stwórz użytkownika [test2](#) i wykonaj podobny test możliwości przesyłania plików na serwer
 - i. Każdy użytkownik po zalogowaniu widzi w swoim [katalogu macierzystym](#) jedynie listę własnych plików i podkatalogów.
- c. Zaloguj się jako [test1](#)
 - i. w [katalogu macierzystym](#) i jego podkatalogach kliknij na ikony i nazwy przesłanych tam wcześniej plików, w celu przetestowania możliwości odtworzenia ich w aplikacji (o ile są wspierane przez przeglądarkę, np. jpg, png, gif, mp3, mp4) oraz ściągnięcia ich z serwera i zapisania ich na lokalnym komputerze.
 - ii. Przetestuj, czy pobrane z serwera pliki są poprawne – tzn. czy podczas przesyłania na serwer nie został uszkodzony ich format i czy wyświetlają się one poprawnie w odpowiednich aplikacjach.
 - iii. Wykonaj te testy rzetelnie, ponieważ wykładowca do testowania tej aplikacji wykorzystuje różne typy plików, aby się przekonać, czy działa ona poprawnie dla dowolnego typu pliku.
- d. Otwórz inną przeglądarkę, niż ta, z której przesyłałeś pliki użytkowników [test1](#) i [test2](#) oraz testowałeś pliki użytkownika [test1](#).
 - i. Zaloguj się jako [test2](#).
 - ii. Przetestuj możliwość odtwarzania i ściągnięcia plików użytkownika [test2](#).
 - iii. Zmiana przeglądarki podczas testowania niektórych aplikacji webowych jest bardzo ważna, ponieważ pozwala wykryć błędy, które mogłyby umknąć z powodu cach'owania plików przez przeglądarkę.

8. Dodatkowe funkcjonalności portalu dostępne po zalogowaniu się użytkownika (zadania dla wskazanych przez wykładowcę grup studentów):

- a. Dodaj funkcję przełączania wyświetlania plików w portalu
 - i. zwykła lista nazw plików,
 - ii. nazwy plików + miniatury dla plików graficznych lub playery filmów lub plików dźwiękowych,
 - iii. jeśli dany plik nie jest obrazkiem, filmem lub plikiem audio – wyświetla się jedynie w postaci nazwy pliku, bez względu na to, w jakim trybie pracuje portal,
 - iv. wyświetlanie oprócz nazw plików również dat ich utworzenia oraz rozmiarów,
- b. tworzenie maksymalnie czterokrotnie zagłębionych struktur podkatalogów wewnątrz [katalogu macierzystego](#),

- c. utworzenie funkcji udostępniania plików i katalogów innym użytkownikom aplikacji, podobne funkcje są dostępne m.in. w poniższych systemach
 - i. <https://support.apple.com/en-us/HT210910>
 - ii. <https://support.google.com/drive/answer/7166529?hl=en&co=GENIE.Platform%3DDesktop>
 - iii. <https://support.microsoft.com/en-us/office/share-onedrive-files-and-folders-9fcc2f7d-de0c-4cec-93b0-a82024800c07>
 - iv. <https://docs.aws.amazon.com/workdocs/latest/userguide/share-docs.html>
- d. utworzenie aplikacji desktopowej w VS2022 umożliwiającej wyświetlenie listy loginów użytkowników portalu.
- e. Prawidłowo zrealizowane aplikacje sieciowe (szczególnie realizowane jako prace inżynierskie lub magisterskie) powinny być dobrze udokumentowane oraz zgodne z wytycznymi dotyczącymi technologii realizacyjnych, dlatego po zrealizowaniu aplikacji należy wykonać poniższe badania:
 - i. Audyt zgodności z technologią HTML 5 oraz CSS3 (należy do tego wykorzystać specjalistyczne portale internetowe lub narzędzia webmasterskie) i umieścić ich raporty w sprawozdaniu.
 - ii. Sprawdzanie, czy nie ma błędów kodu w portalu (j.w.).
 - iii. Sprawdzanie, czy portal nie wykazuje podatności na włamania typu SQL injection i in (j.w.).
 - iv. Wykonanie przykładowego automatycznego testu interfejsu użytkownika z wykorzystaniem bezpłatnego lub w pewnych warunkach bezpłatnego specjalistycznego narzędzia np. <http://www.seleniumhq.org/>.
 - v. Sprawdzenie szybkości ładowania strony, ewentualna optymalizacja zastosowanej grafiki i ponowne sprawdzenie szybkości ładowania strony (optymalizacja portalu pod względem szybkości ładowania).
 - vi. Generowanie mapy portalu (należy do tego wykorzystać specjalistyczne portale internetowe).
 - vii. Próba ściągnięcia całego portalu za pomocą narzędzi do pobierania statycznych wersji serwisów internetowych (np. Teleport Pro). Po ściągnięciu zawartości testowanego portalu na dysk należy sprawdzić, czy aplikacji udało się zaciągnąć jako wartość offline jakieś elementy, których nie życzilibyśmy sobie udostępniać osobom niezalogowanym, bo np. pogarsza to bezpieczeństwo portalu.
 - viii. Dokumentacja BD MySQL – zastosowanych w aplikacji tabel i relacji z wykorzystaniem specjalistycznych narzędzi np. <http://dbdesc.com>, Workbench, <http://www.dbschema.com/>