

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Шеремет Георгій Олександрович,
студент групи РФ-171

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Організація дистанційного доступу до комп'ютерної навчальної
лабораторії за допомогою web-технологій

Спеціальність:

122 Комп'ютерні науки

Спеціалізація, освітня програма:

Комп'ютерні науки та інформаційна безпека

Керівник:

Стопакевич Олексій Аркадійович,

к.т.н., доц.

Одеса – 2022

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
Спеціалізація, освітня програма
Комп'ютерні науки та інформаційна безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ

д.т.н., проф. А.А.Кобозева
_____ 2022р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Шеремету Георгію Олександровичу

1. Тема роботи: *Організація дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою web-технологій,*
керівник роботи *Стопакевич Олексій Аркадійович, к.т.н., доц.,*
затверджені наказом ректора від „_____” _____ 20__ р. № _____ .
2. Зміст роботи: *аналіз проблемної області, постановка задачі, розробка проекту системи дистанційного доступу до комп'ютерної навчальної лабораторії, програмна реалізація системи дистанційного доступу до комп'ютерної навчальної лабораторії.*
3. Перелік ілюстративного матеріалу: *слайди презентації.*
4. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Дата видачі завдання “ _____ ” _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	<i>Аналіз джерел з теми випускної кваліфікаційної роботи</i>	25.10.2022	<i>виконано</i>
2	<i>Обґрунтування вибору рішення. Збір даних</i>	01-11-2022	<i>виконано</i>
3	<i>Аналіз основних аспектів організації дистанційного доступу</i>	05-11-2022	<i>виконано</i>
4	<i>Розроблення проекту системи дистанційного доступу</i>	09-11-2022	<i>виконано</i>
5	<i>Програмна реалізація системи дистанційного доступу</i>	22-11-2022	<i>виконано</i>
6	<i>Підготовка тексту роботи</i>	25-11-2022	<i>виконано</i>
7	<i>Підготовка презентації та доповіді</i>	28-11-2022	<i>виконано</i>
8	<i>Попередній захист</i>	02-12-2022	<i>виконано</i>
9	<i>Нормоконтроль, рецензування</i>	17-12-2022	<i>виконано</i>
10	<i>Занесення роботи в електронний архів</i>	23-12-2022	<i>виконано</i>
11	<i>Допуск до захисту у завідувача кафедри</i>	23-12-2022	<i>виконано</i>

Здобувач вищої освіти

Шеремет Г.О.

Керівник роботи

Стопакевич О.А.

АНОТАЦІЯ

Кваліфікаційна робота на тему “ Організація дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою web-технологій” на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 122 Комп'ютерні науки, спеціалізація, освітня програма: Комп'ютерні науки та інформаційна безпека, містить 38 рисунків, 1 додаток, 29 літературних джерел за переліком посилань. Робота виконана на 56 сторінках загального тексту і 52 сторінках основного тексту.

Мета роботи – організація та автоматизація дистанційного доступу до комп'ютерної навчальної лабораторії.

У роботі організовано дистанційний доступ до комп'ютерної навчальної лабораторії за допомогою web-технологій та автоматизований почерговий доступ студентів до лабораторії.

У результаті виконання кваліфікаційної роботи розроблено систему дистанційного доступу до комп'ютерної навчальної лабораторії, яка має систему реєстрації, автентифікації та авторизацію користувачів, надає користувачу можливість віддаленого керування курсором та клавіатурою, надає користувачу можливість віддаленого запуску програмного забезпечення, надає можливість обміну файлами між персональним комп'ютером користувача та комп'ютером навчальної лабораторії, забезпечує почергове використання комп'ютерів користувачами. Впровадження системи дистанційного доступу дозволяє використовувати навчальну лабораторію дистанційно.

Результати даної роботи можуть бути використані при побудові аналогічної системи дистанційного доступу.

ДИСТАНЦІЙНИЙ ДОСТУП, ДИСТАНЦІЙНИЙ КОВОРКІНГ, SSH, БАЗА ДАНИХ.

ANNOTATION

Qualification work on the topic "Organization of remote access to a computer educational laboratory using web technologies" for obtaining the second (master's) level of higher education in the specialty 122 Computer science, specialization, educational program: Computer science and information security, contains 38 figures, 1 appendix, 29 literary sources according to the list of references. The work is completed on 56 pages of the general text and 52 pages of the main text.

The purpose of the work is the organization and automation of remote access to the computer training laboratory.

The work organizes remote access to the computer educational laboratory using web technologies and automated alternate access of students to the laboratory.

As a result of the qualification work, a system of remote access to the computer training laboratory was developed, which has a system of user registration, authentication and authorization, provides the user with the ability to remotely control the cursor and keyboard, provides the user with the ability to remotely launch software, provides the ability to exchange files between personal computers the user's computer and the computer of the educational laboratory, ensures alternate use of computers by users. Implementation of the remote access system allows you to use the educational laboratory remotely.

The results of this work can be used in the construction of a similar remote access system.

REMOTE ACCESS, REMOTE CO-WORKING, SSH, DATABASE.

ЗМІСТ

Вступ.....	8
1 Аналіз проблеми дистанційного коворкінга та навчання.....	10
1.1 Основні проблеми дистанційного коворкінга та навчання.....	10
1.2 Аналіз типових рішень дистанційного коворкінга	12
1.3 Огляд сучасних елементів систем дистанційного доступу до комп'ютерних мереж.....	15
2 Проектування системи дистанційного доступу до комп'ютерної навчальної лабораторії.....	24
2.1 Загальна схема роботи системи.	24
2.2 Проектування бази даних.	25
2.3 Життєвий цикл сесії підключення до віддаленого комп'ютера.....	27
2.4 Архітектурний патерн MVC.....	28
2.5 UML діаграма класів.	30
3 Програмна реалізація системи дистанційного доступу до комп'ютерної навчальної лабораторії.....	33
3.1 Середовище програмування.....	33
3.1.1 Операційна система Ubuntu	33
3.1.2 Мова програмування PHP	34
3.1.3 База даних MySQL	36
3.1.4 Web-сервер Nginx.....	37
3.1.5 Laravel фреймворк.....	38
3.1.6 Bootstrap	40
3.1.7 Docker та Docker Compose	41
3.2 Розгортання системи	42
3.2.1 Встановлення RustDesk.	43
3.2.1 Встановлення та налаштування пакета виконуваних файлів.	45
3.2.2 Встановлення сервера OpenSSH.....	46
3.2.3 Створення задачі в Windows.....	48
3.3 Моделювання роботи системи	49
3.3.1 Панель адміністратора.....	49
3.3.2 Реєстрація та аутентифікація в системі	50

3.3.3	Підключення до комп'ютера віддаленої навчальної лабораторії	50
3.3.4	Черга на підключення.....	52
	Висновки	54
	Перелік посилань.....	56

ВСТУП

Зі зростанням кількості віддалених робочих місць все більше людей виявляють, що можуть працювати без обмежень, пов'язаних з офісною роботою з 9 до 5. Незважаючи на те, що віддалена робота має безліч переваг, таких як можливість гнучкого складання власного графіка, одна річ, яку цей стиль роботи не може врахувати, це зв'язок і взаємодія з людьми, особливо якщо ви працюєте з дому. Ці труднощі допомагає вирішити коворкінг.

Коворкінг - це модель, в якій люди збираються разом, щоб працювати незалежно чи спільно у спільному просторі. Як правило, це фізичний простір, такий як коворкінг, проте віртуальні або онлайн-коворкінги є тенденцією, що росте, особливо через збільшення числа людей, що працюють вдома під час пандемії Covid-19.

Структурні зміни у світовій системі освіти, що відбулися починаючи з другій половині ХХ століття, зумовлені розвитком науково-технічного прогресу, вплинули на всі сторони життя суспільства. Поява дистанційного навчання не є раптовою подією, за всіх часів потреба у освіті зберігалася високому рівні. Поява інтернету та прискорення темпів наукового прогресу лише сприяли поширення даного формату здобуття освіти. У світлі останніх подій в Україні дистанційне навчання набуває ще більшої актуальності.

Існує безліч готових рішень для віддаленого доступу до комп'ютера. Використання найбільш функціональних з них є дуже дорогим, особливо для дистанційної освіти. Для вирішення цієї проблеми була розроблена дана система, хай і обмежено, але безкоштовно вона вирішує основні завдання. У роботі розглядається організація дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою web-технологій з використанням виключно безкоштовних рішень. При цьому запропоноване рішення є досить функціональним і забезпечує студентів всім необхідним для комфортного використання навчальної лабораторії дистанційно.

Мета роботи: організація та автоматизація дистанційного доступу до комп'ютерної навчальної лабораторії.

Для досягненні поставленої мети в роботі розв'язувалися наступні задачі:

- розробити систему реєстрації, автентифікації та авторизацію користувачів;
- надати можливість віддаленого керування курсором та клавіатурою;
- надати можливість віддаленого запуску програмного забезпечення;
- надати можливість обміну файлами між персональним комп'ютером користувача та комп'ютером навчальної лабораторії;
- забезпечити по чергове використання комп'ютерів студентами.

Матеріали кваліфікаційної роботи знайшли своє відображення в статті «Організація дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою веб-технологій» [1].

1 АНАЛІЗ ПРОБЛЕМИ ДИСТАНЦІЙНОГО КОВОРКІНГА ТА НАВЧАННЯ

1.1 Основні проблеми дистанційного коворкінга та навчання

Коворкінг означає роботу двох або більше людей в одному місці, але не в одній фірмі. Це явище набуло значного розмаху в останні роки, включаючи зростання попиту у великих містах навколо світу та перспективи подальшого зростання в майбутньому. Характеристика фірм, розташованих у коворкінгах здається, змінилися з часом, оскільки сьогодні також зростає кількість великих нетехнологічних фірм, які обирають цей стиль [2].

У цілому нині з підприємницького погляду перебування у коворкінгу позитивно впливає на підприємницьку поведінку. Це засноване в основному на позитивних побічних ефектах та співпраці представників різних професій у рамках коворкінгу, що особливо вигідно молодим фірмам [2,3,4].

Коворкінг явище, яке проникло в організаційні структури, впливає на створення та обмін знаннями, покращує інноваційну поведінку та перебуває під впливом соціальних факторів, а також матеріального оснащення. Середовище, що нагадує коворкінг, робить людей більш щасливими та емоційно здоровими, компанії можуть розглянути можливість застосування набутого досвіду та знань у своїй організації, щоб співробітники з більшою готовністю залишалися з ними [5,6,7].

Дистанційне навчання дозволяє студентам отримувати доступ до якісного освітнього контенту та працювати з ним, не виходячи з дому [8].

Хоча легко побачити як переваги, так і актуальність дистанційної освіти, такі як підвищена гнучкість, мобільність і доступність, не менш важливо визнати потенційні проблеми дистанційного навчання для студентів. Від учнів початкових класів до аспірантів, учні на всіх вікових рівнях повинні долати перешкоди, які є унікальними для віртуальної освіти, такі як асинхронне спілкування та постійні можливості для відволікання, що витрачає час. Учні також стикаються з проблемами, які, хоча часто присутні

в традиційному класі, можуть бути посилені у віртуальному середовищі, наприклад, із затримкою або нечітким відгуком інструктора.

Електронне навчання може створити певні труднощі для студентів, воно також створює захоплюючі освітні можливості, відкриваючи двері як для молодих, так і для дорослих учнів.

Ось чотири недоліки віртуального навчання.

Неефективне управління часом. Дослідження довели, що успішне «розпорядження часом пов'язане з вищою академічною успішністю та нижчим рівнем тривоги у студентів». На жаль, це ж дослідження також виявило, що «багатьом студентам важко знайти баланс між навчанням і повсякденним життям» — і що ще гірше, що неефективне управління часом було пов'язане з такими результатами, як «поганий сон» і «підвищений рівень стресу». Ефективне управління часом може бути особливо складним у середовищі дистанційного навчання, де учням доводиться самотійно контролювати темп.

Відсутність миттєвого спілкування В особистій обстановці спілкування відбувається миттєво, що дозволяє студентам легко отримувати відповіді та роз'яснювати непорозуміння. В умовах електронного навчання спілкування часто є асинхронним, що означає, що між викладачем і студентом існує розрив. У цих прогалинах легко виникнути непорозуміння — інколи проблема стає сніжним комом, перш ніж її можна буде виправити.

Не отримуєте своєчасний відгук. Надання зворотного зв'язку є одним із найважливіших і значущих способів спілкування вчителя зі студентом. Коли зворотний зв'язок затримується на додаткові дні чи тижні через онлайн-формат, учні можуть збентежитись або бути невпевненими щодо ваших очікувань, свого прогресу та успішності у вашому класі.

Неотримання чітких інструкцій чи очікувань. Завжди вкрай важливо встановлювати чіткі очікування для студентів. В іншому випадку вони можуть лише здогадуватися, чи правильно вони виконують завдання та

проекти. Хоча встановлення чітких стандартів є проблемою в будь-якому класі, асинхронне спілкування може зробити це ще більшою перешкодою.

1.2 Аналіз типових рішень дистанційного коворкінга

Детальна інформація про технологію VPN Always ON [9,10].

Організація безпечного віддаленого доступу є однією з ключових складових віддаленої роботи та розширення можливостей мобільного робочого місця. Always On VPN допомагає забезпечити зашифрований та безпечний приватний канал за допомогою мережі Інтернет з будь-якого місця до корпоративних даних та додатків. ІТ-персонал може призначати права доступу різних груп користувачів на основі списків доступу Active Directory. Архітектура Always On VPN зображено на рисунку 1.1.

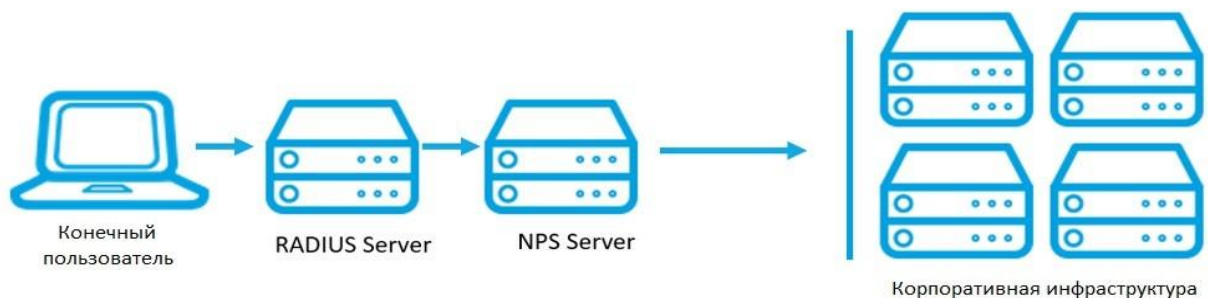


Рисунок 1.1 – Архітектура Always On VPN

Переваги:

- Легке розгортання;
- Економічна ефективність;
- Відповідність вимогам безпеки.
- Необхідно мати:
- Windows Server;
- Операційна система Windows 10;
- Не потрібна додаткова інфраструктура;

Використання та інтеграція здійснюється одноразово без регулярних платежів;

- Простота в управлінні;
- Архітектура відповідає вимогам безпеки Microsoft;
- Захищено мультифакторною автентифікацією.

Детальна інформація про технологію Windows Virtual Desktop (за допомогою MS Azure) [11].

Комплексне хмарне рішення для віртуалізації робочих місць, столів та додатків. Архітектура Windows Virtual Desktop зображено на рисунку 1.2.

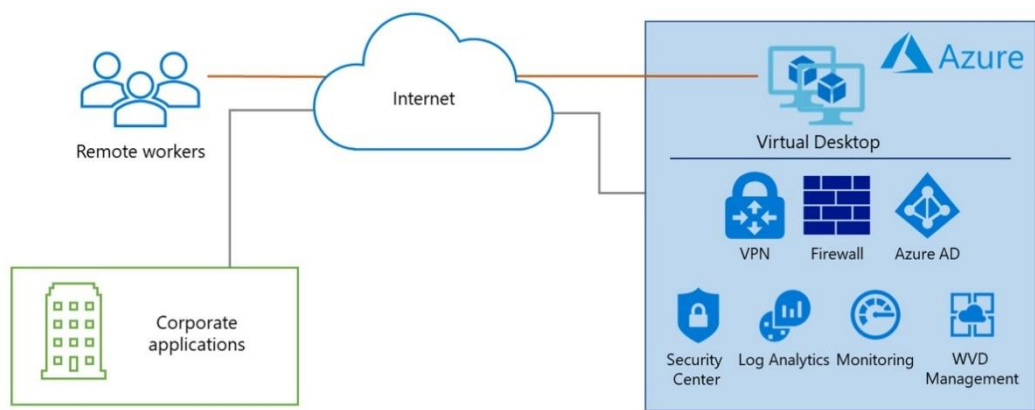


Рисунок 1.2 – Архітектура Windows Virtual Desktop

Переваги:

- швидке розгортання і легка неограничена масштабованість;
- спрощене керування;
- багатосесійний, односесійний режим Windows 10;
- режим публікації програм;
- вбудовані функції забезпечення безпеки та відповідності;
- авторизація за допомогою Active Directory та/або Azure Active Directory.

Directory.Azure AD Application Proxy [12].

Проксі AD надає єдиний вхід (SSO) і безпечний віддалений доступ для веб-додатків, розміщених локально.

Деякі програми, які потрібно опублікувати, можуть включати вузли SharePoint, Outlook Web Access або будь-які інші бізнес-програми, що використовуються.

Ці локальні веб-програми середовища інтегруються з платформою ідентифікації та керування Azure AD, яка використовується і для доступу до Office 365.

Користувачі можуть звертатися до локальних програм так само, як до програм Office 365 та інших пропозицій SaaS, інтегрованих з Azure AD. Для цього не потрібно створювати VPN або змінювати мережну інфраструктуру. Архітектура Azure AD Application Proxy зображено на рисунку 1.3.

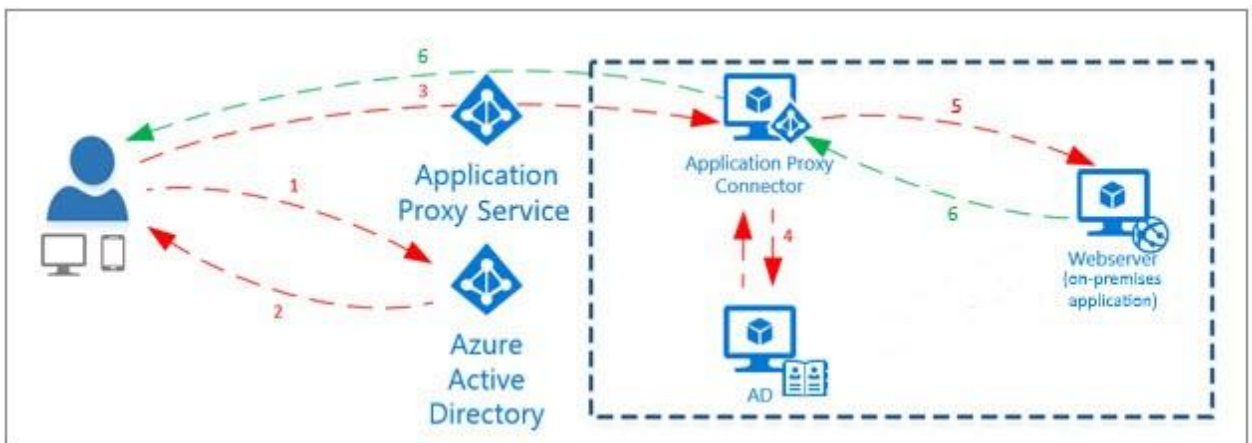


Рисунок 1.3 – Архітектура Azure AD Application Proxy

Особливості:

- користувачі отримують єдиний інтерфейс автентифікації;
- немає необхідності змінювати або оновлювати програми для роботи з проксі-програмами;
- при публікації програми за допомогою проксі програми Azure AD вам доступні засоби аналітики системи безпеки та елементи керування авторизацією, що надаються Azure
- немає потреби у вхідних з'єднаннях через брандмауер, щоб надати користувачам віддалений доступ.

– проксі-додатки виконується у хмарі, дозволяючи заощаджувати час та гроші в порівнянні з наземними рішеннями.

Remote Desktop Service (RDS) у локальній інфраструктурі [13]. Remote desktop Service (у MS Server 2008 і раніше - служба терміналів) – компонент Microsoft Windows Server, що дозволяє користувачеві у віддаленому режимі керувати комп'ютером або віртуальною машиною мережного з'єднання за допомогою протоколу віддаленого робочого столу (Remote Desktop Protocol).

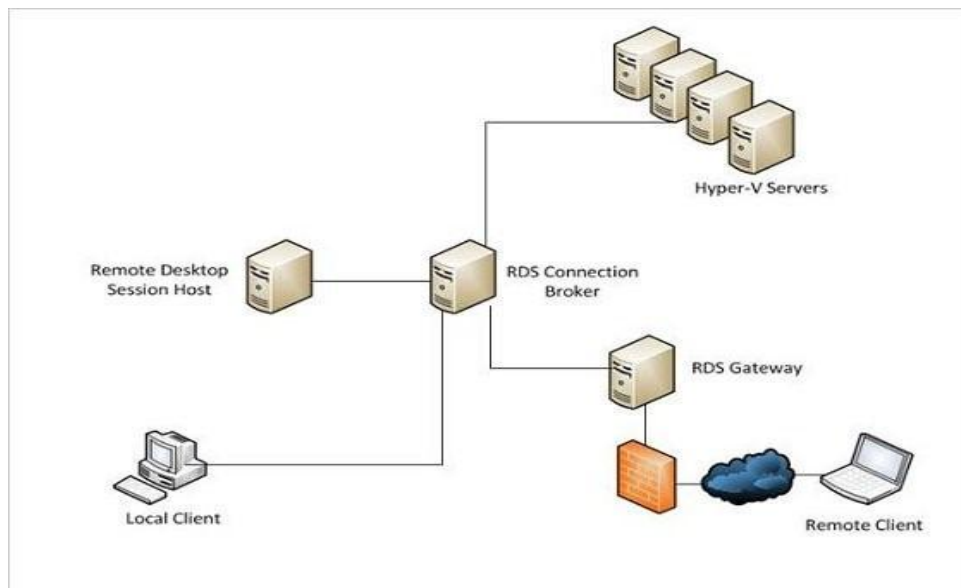


Рисунок1.4 – Архітектура Remote Desktop Service

Особливості: використання меншої кількості Windows, відповідно сховище завантажено менше; при даному сценарії (на відміну від VDI) менше завантажено центральний процесор і пам'ять, відповідно з'являється можливість підключення більшої кількості користувачів.

1.3 Огляд сучасних елементів систем дистанційного доступу до комп'ютерних мереж

В рамках огляду представлено сім програмних розробок, які мають гідний функціонал, пропонують ефективні та дієві інструменти. В більшості випадків йдеться про безкоштовні програми для віддаленого доступу, що також важливо.

1. TrustViewer [14].

TrustViewer розповсюджується безкоштовно. Інтерфейс програми TrustViewer зображено на рисунку 1.5.

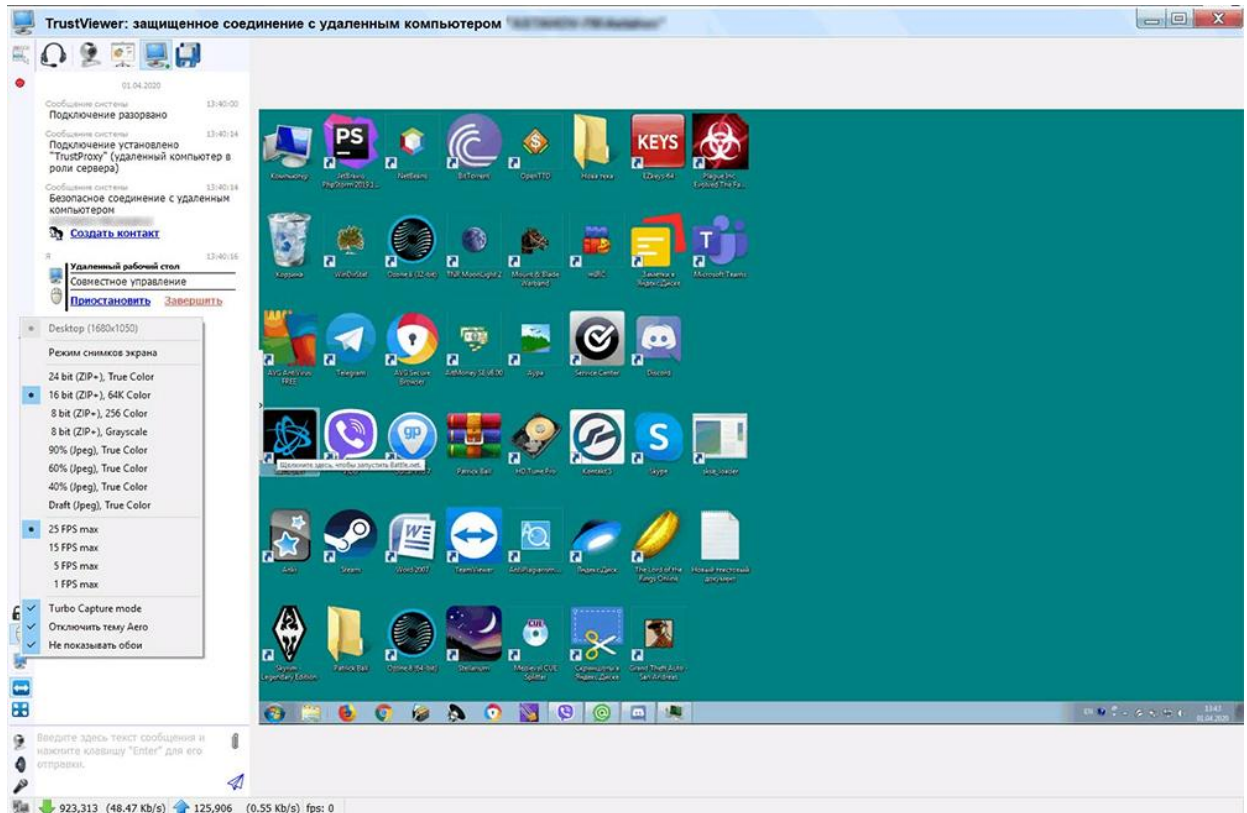


Рисунок 1.5 – Інтерфейс програми TrustViewer

Серед особливостей можна відзначити:

- наявність російськомовного меню;
- присутній вбудований файловий менеджер;
- немає обмежень щодо часу роботи;
- проста у використанні;
- є вбудований відео чат та голосовий зв'язок;
- працює без встановлення на жорсткий диск;
- ніяких особистих даних при першому запуску вводити не потрібно.

До недоліків можна віднести відсутність мобільного клієнта, а також програма не підтримує підключення через проксі-сервер. В іншому відмінна та легка в роботі для новачків утиліта.

З'єднання між комп'ютерами здійснюється за допомогою згенерованого коду доступу. Його потрібно надіслати користувачеві другого комп'ютера, і цим встановиться зв'язок.

2. LiteManager [15].

Вибираючи хороші програми, призначені для отримання віддаленого доступу до іншого комп'ютера безпосередньо через інтернет, варто включити до цього списку LiteManager. Інтерфейс програми LiteManager зображено на рисунку 1.6.

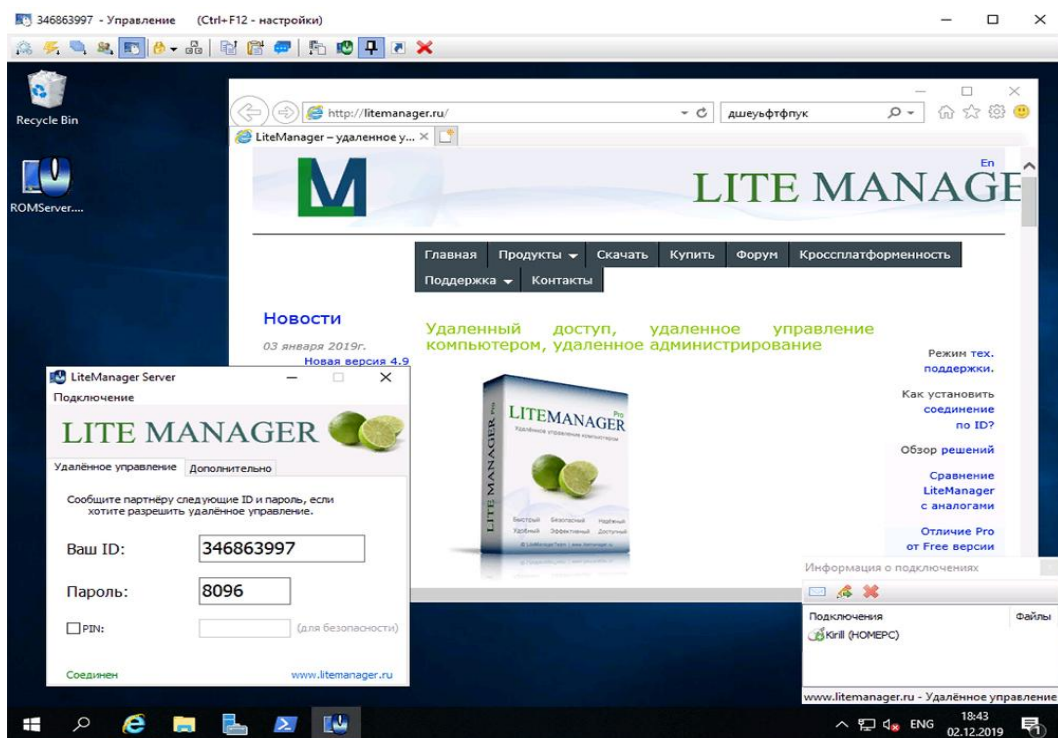


Рисунок 1.6 – Інтерфейс програми LiteManager

Як і в попереднього варіанта, мобільного клієнта цього ПЗ немає. Програма досить складна в роботі, але має широкі можливості. Серед них варто виділити такі:

- можна працювати не більше ніж із 30 комп'ютерами абсолютно безкоштовно;
- тут є інтегрований файловий менеджер;
- є редактор реєстру;
- є свій чат та диспетчер завдань;

- можна записувати те, що відбувається на екрані;
- немає обмежень щодо часу роботи;
- віддалено можна встановлювати та налаштовувати програмне забезпечення.

Щоб ця програма для отримання віддаленого доступу працювала, тобто була можливість підключитися до ПК, потрібно завантажити 2 пакети. Перший називається Server і ставиться на клієнтський комп'ютер. Другий пакет Viewer і його потрібно встановити на ПК оператора.

У випадку клієнта в управлінні немає нічого складного. А ось оператор спочатку має розібратися з усім доступним функціоналом.

3. Remote Utilities [16].

Це одна з найфункціональніших програм, що потрапили в топ серед найкращих рішень для віддаленого доступу. Розробником виступає компанія Usoris. Інтерфейс програми Remote Utilities зображено на рисунку 1.7.

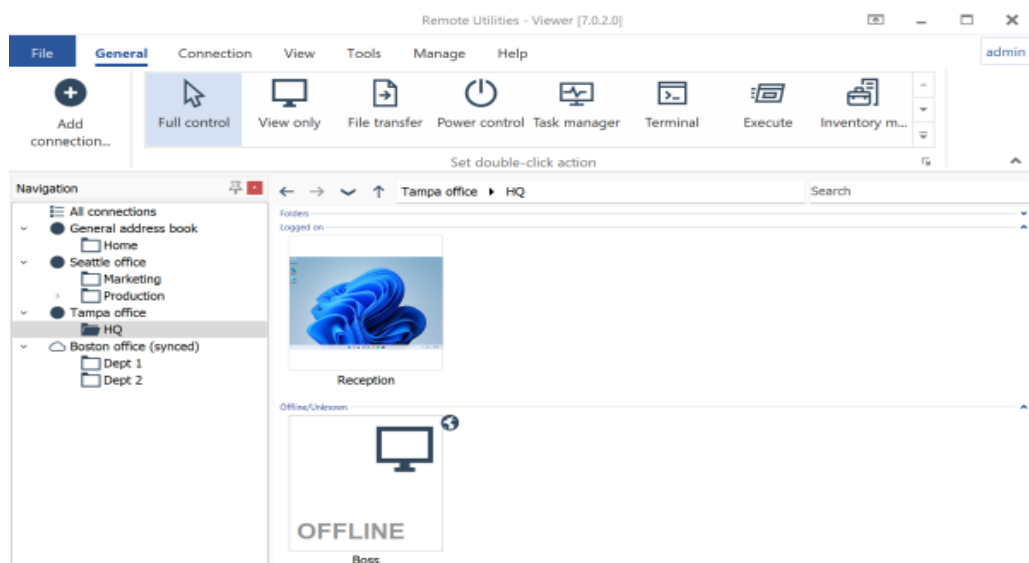


Рисунок 1.7 – Інтерфейс програми Remote Utilities

Якщо говорити про слабкі сторони софту, то сюди можна віднести лише часткову русифікацію меню, порівняно складне використання та можливість завантаження лише на 3 останні версії ОС Windows. Тобто 7, 8 та Windows 10.

Але є й низка переваг, завдяки чому софт і потрапив до рейтингу серед найкращих програм для організації віддаленого доступу до комп'ютерів.

Сильні сторони такі:

- доступно відразу кілька режимів з'єднання;
- можна віддалено встановлювати та розгортати програмне забезпечення;
- є функція запису екрана;
- є режим мультимонітор, відображаючи кілька моніторів віддалених ПК одночасно;
- програма має портативну версію;
- у софт інтегровані відео, аудіо та текстовий чат;
- є своя файлова система;
- розроблені мобільні версії для Android та iOS.

Також ця розробка є безкоштовною. Якщо бути точнішим, у безкоштовній версії є обмеження на роботу з 10 комп'ютерами, не більше.

4. Remote Desktop від Chrome [17].

Якщо говорити про кращі програми, створені для організації дистанційного керування іншим комп'ютером, то без розробки Chrome список був би неповним. У цієї утиліти свої особливості.

Варто зазначити, що у плані функціоналу софт дещо обмежений і не дає стільки ж можливостей, як деякі просунуті спеціалізовані програми. Тобто вам потрібна проста та безпечна програма для реалізації віддаленого керування своїм чи чужим робочим столом, це буде чудовий варіант.

Ще одним умовним недоліком є обов'язкове встановлення браузера Chrome. Інакше софт не працюватиме. Інтерфейс програми Remote Desktop зображено на рисунку 1.8.

Зараз до переваг. Вони такі:

- високий рівень безпеки під час віддаленої роботи на іншому ПК;

- не потрібно встановлювати зайве ПЗ, щоб швидко виконати підключення;
- поширюється абсолютно безкоштовно;
- може працювати на будь-якому комп'ютері та девайсі, де є Chrome.

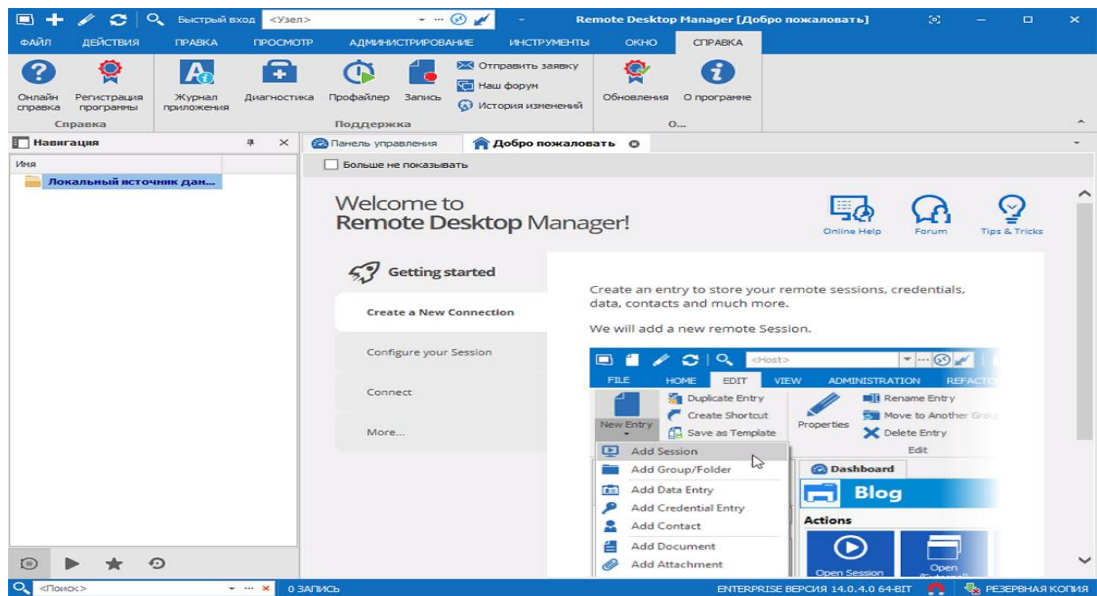


Рисунок 1.8 – Інтерфейс програми Remote Desktop

Відмінною особливістю цієї утиліти є те, що керування здійснюється через браузер. Причому утиліта відкриває доступ не тільки до веб-браузера іншого ПК, але й до всього робочого столу.

5. TeamViewer [18].

Для управління чужими комп'ютерами та організації віддаленого доступу до них добрим рішенням може стати програма TeamViewer.

Якщо говорити про домашнє користування, то для таких випадків є безкоштовна версія. Так, функціонал буде обмежений, але для вирішення повсякденних стандартних завдань його цілком достатньо. Інтерфейс програми TeamViewer зображено на рисунку 1.9.

До сильних сторін можна віднести:

- високий рівень захисту даних;
- є безкоштовна версія;

- щоб прискорити передачу файлів вони автоматично стискаються;
- у програмі видно, у мережі користувач чи ні;
- меню російськомовне;
- на неї не реагує брандмаузер, даючи спокійно працювати;
- присутній вбудований чат та файлообмінник;
- доступні мобільні версії для Windows, Android та iOS;
- проста у використанні програма;
- хороша функціональність.

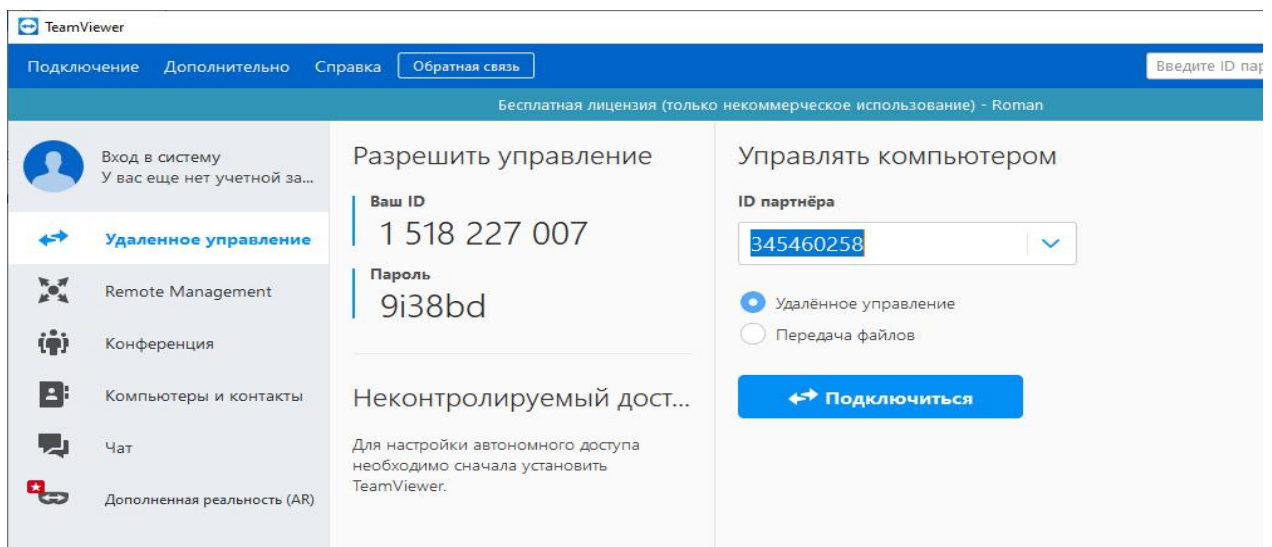


Рисунок 1.9 – Інтерфейс програми TeamViewer

Дійсно відмінна, легка і проста у розумінні програма, що дає досить широкі можливості для користувачів. Тому входить до найпопулярніших програмних розробок свого сегмента.

6. UltraVNC [19].

Приклад ще однієї програми, що добре себе зарекомендувала, для організації віддаленого доступу для управління своїм або чужим комп'ютером.

Варто одразу виділити недоліки. Основний мінус у тому, що програма повністю виконана англійською мовою. Плюс первинне з'єднання викликає певні складності, якщо не задіяти спеціальні розширення. Інтерфейс програми UltraVNC зображено на рисунку 1.10.

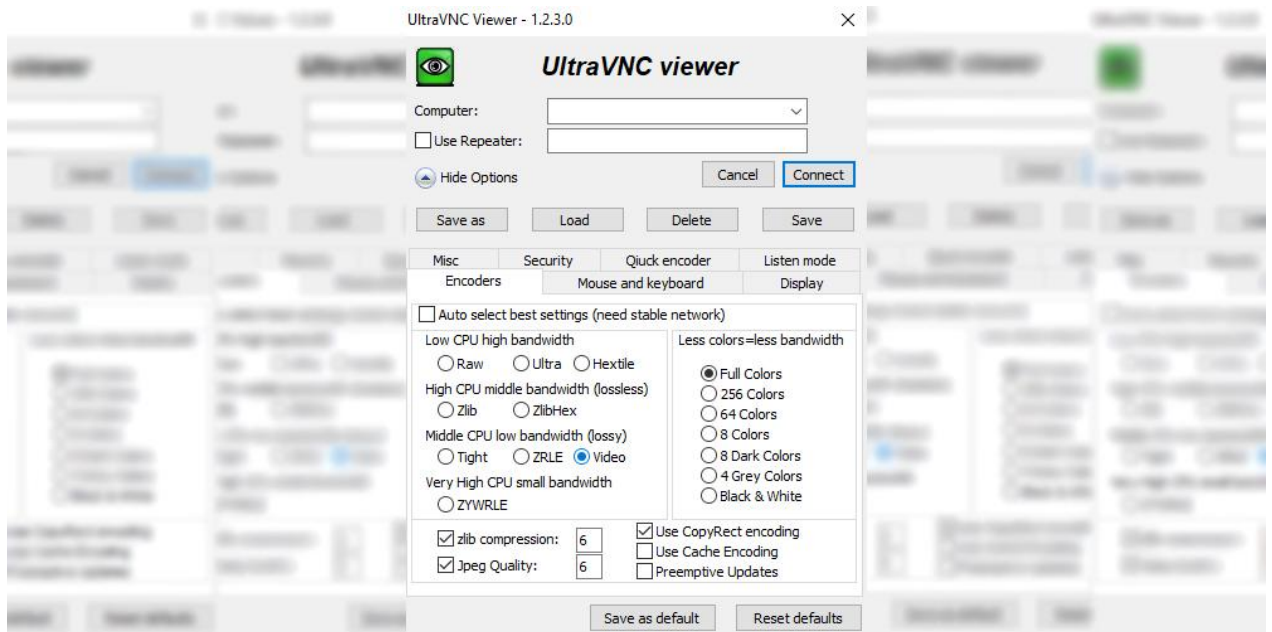


Рисунок 1.10 – Інтерфейс програми UltraVNC

Що ж до переваг, то тут варто виділити такі пункти:

- утиліта поширюється абсолютно безкоштовно;
- є вбудований чат;
- є інтегровані плагіни;
- зображення автоматично масштабуються;
- доступні мобільні версії під різні платформи;
- підтримує проксі-сервер;
- дані, що передаються, шифруються.

Утиліта відкриває досить широкі можливості. Зручний інструмент для тих, хто хоче створити мережу VPN і підключатися до віддалених робочих столів.

7. RustDesk [20].

Підтримувані платформи: Windows, MacOS, Linux, Android, iOS.

Вартість: безкоштовно (Open Source). Інтерфейс програми RustDesk зображено на рисунку 1.11.

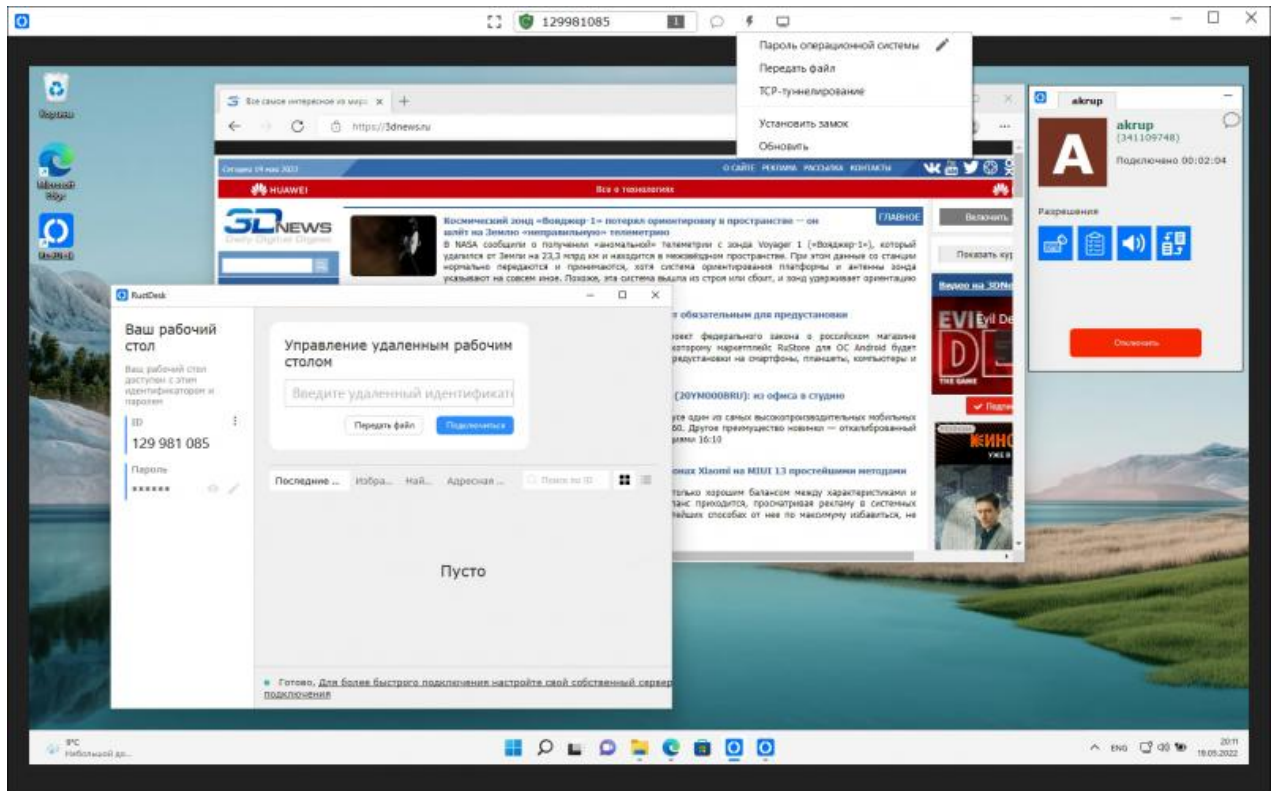


Рисунок 1.11 – Интерфейс програми RustDesk

Продукт із відкритим вихідним кодом (опубліковано на GitHub). Для прокладки мережних маршрутів між віддаленими пристроями використовує власний публічний сервер, вихідні та інсталяційні пакети якого також є у відкритому доступі. RustDesk простий у налаштуванні (для хоста та клієнта використовується одна програма), підтримує обмін повідомленнями та файлами, дозволяє налаштовувати права доступу. Також є веб-клієнт, що дозволяє підключатися до віддалених пристроїв та керувати ними за допомогою браузера. Програма розповсюджується розробниками як вигляді інсталяційних дистрибутивів, так і у форматі не потребують установки портативних збірок.

2 ПРОЕКТУВАННЯ СИСТЕМИ ДИСТАНЦІЙНОГО ДОСТУПУ ДО КОМП'ЮТЕРНОЇ НАВЧАЛЬНОЇ ЛАБОРАТОРІЇ

На основі аналізу програм дистанційного доступу, проведеного в попередньому розділі, було зроблено висновок, що стандартного функціоналу програм дистанційного доступу недостатньо для досягнення поставлених у роботі завдань. Отже необхідно доопрацювати недостатній функціонал.

Було ухвалено рішення взяти програму RustDesk за основу. Обрана програма повною мірою вирішує наступні задачі: надати можливість віддаленого керування курсором та клавіатурою; надати можливість віддаленого запуску програмного забезпечення; надати можливість обміну файлами між персональним комп'ютером користувача та комп'ютером навчальної лабораторії. Відштовхуючись від цього, було виконано подальше проектування системи.

2.1 Загальна схема роботи системи

Запропонована наступна загальна схема організації дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою web-технологій, яка має вигляд, представлений на рис. 2.1.

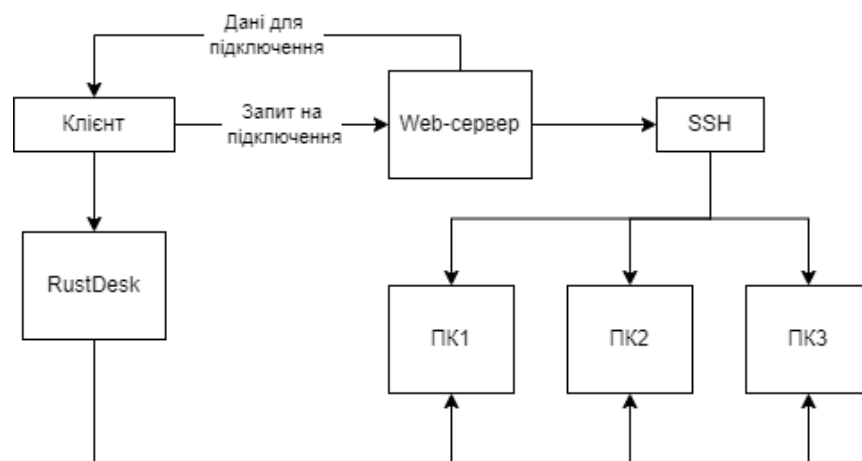


Рисунок 2.1 – Загальна схема роботи системи

де: Клієнт – це користувач системи; Web-сервер - це сервер, який приймає запити від клієнтів і видає їм відповіді; SSH – це мережевий протокол прикладного рівня, що дозволяє здійснювати віддалене керування операційною системою; RustDesk – програма дистанційного доступу до комп'ютера; ПК1, ПК2, ПК3 – це персональні комп'ютери навчальної лабораторії.

Дистанційний доступ організовано так.

1. Клієнт надсилає запит на підключення веб-серверу.
2. Web-сервер підключається до ПК за протоколом SSH та встановлює пароль для підключення.
3. Web-сервер повертає клієнту відповідь, що містить ID та пароль.
4. Клієнт підключається до віддаленого комп'ютера навчальної лабораторії через програму RustDesk, надаючи ID та пароль раніше отримані від Web-сервера.
5. Web-сервер перериває дистанційний доступ після закінчення відведеного часу (наприклад через одну годину).

У цьому розділі було розглянуто загальну схему роботи системи, в наступних розділах ми розглянемо роботу системи більш детально.

2.2 Проектування бази даних

Схема бази даних розробленої системи представлена на рисунку 2.2.

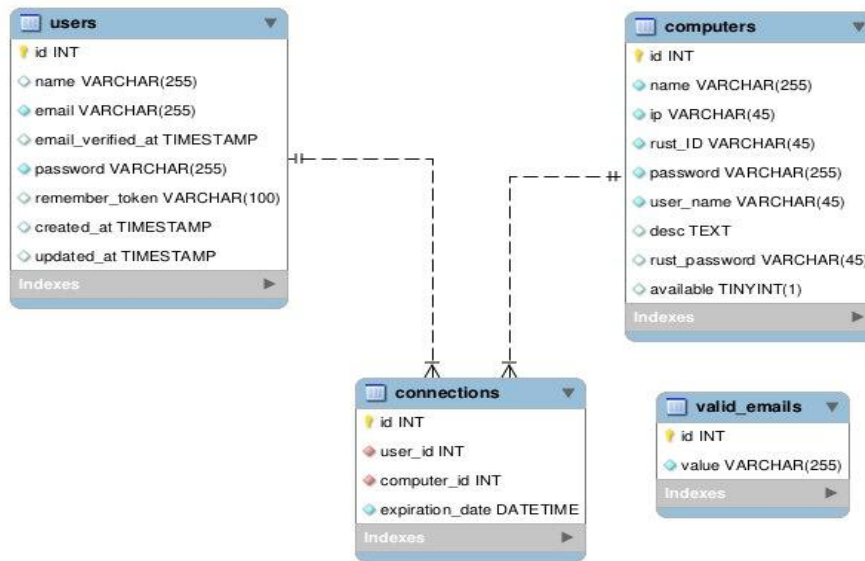


Рисунок 2.2 – Схема бази даних

Розглянемо спроектовану базу даних докладніше.

Таблиця user має такі поля:

- id – ідентифікатор;
- name – ім'я користувача;
- email – електронна адреса користувача, унікальна.
- email_verified_at – дата верифікації електронної адреси користувача.
- password – пароль користувача;
- remember_token - токен для запам'ятовування користувача в системі;
- created_at – дата створення запису;
- updated_at – дата останнього редагування запису.

Таблиця computers має такі поля:

- id – ідентифікатор;
- name – ім'я комп'ютера;
- ip – ip-адреса комп'ютера;
- rust_ID – ID встановленого на комп'ютері RustDesk;
- password – пароль облікового запису на комп'ютері;

- user_name – ім'я облікового запису на комп'ютері;
- desc – опис комп'ютера;
- rust_password – пароль для RustDesk, встановленого на комп'ютері;
- available – стан доступності комп'ютера для підключень;

Таблиця connections має такі поля:

- id – ідентифікатор;
- user_id – ідентифікатор користувача. Зовнішній ключ посилається на поле id у батьківській таблиці user;
- computer_id – ідентифікатор комп'ютера. Зовнішній ключ посилається на поле id у батьківській таблиці computers;
- expiration_date – дата закінчення підключення.

Таблиця valid_emails має такі поля:

- id – ідентифікатор;
- value – домен електронної пошти, доступний для реєстрації.

2.3 Життєвий цикл сесії підключення до віддаленого комп'ютера

Сесія підключення до віддаленого комп'ютера виникає відразу після того, як користувач натисне кнопку «Підключитися» на веб-сторінці. Розглянемо докладніше життєвий цикл сесії підключення. Її появу, зміну стану залежно від зайнятості вибраного комп'ютера та завершення сесії.

На рисунку 2.3 представлений життєвий цикл сесії підключення у разі коли комп'ютер вільний.



Рисунок 2.3 – Життєвий цикл сесії підключення при вільному комп'ютері

Як видно на малюнку, користувач ініціює запит на підключення. Система у свою чергу створює підключення і стежить за тим, щоб через певний час з'єднання було розірвано. Тривалість сесії підключення задає адміністратор системи.

На рисунку 2.4 представлений життєвий цикл сесії підключення якщо комп'ютер зайнятий.

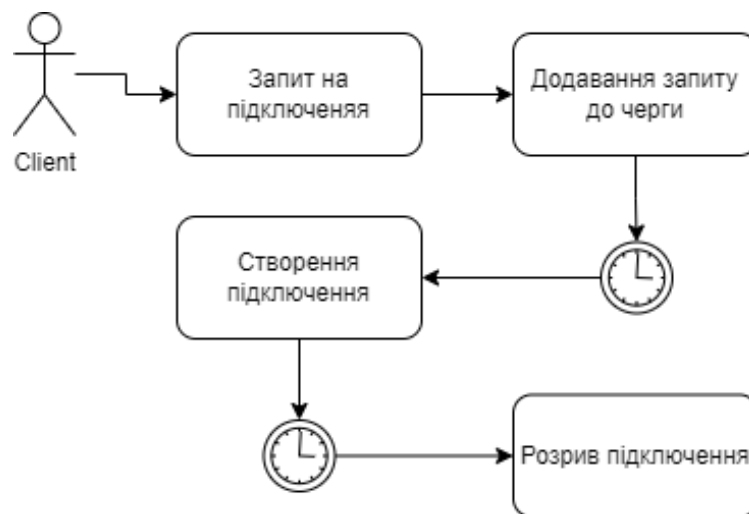


Рисунок 2.4 – Життєвий цикл сесії підключення при зайнятому комп'ютері

На відміну від випадку з вільним комп'ютером, система спочатку додає запит на підключення до черги. Коли прийде черга цього запиту, система ініціює створення підключення. Далі так само як і в попередньому випадку система розірве з'єднання після закінчення часу сесії.

2.4 Архітектурний патерн MVC

При проектуванні системи віддаленого доступу до комп'ютерів цифрової лабораторії було прийнято рішення використовувати архітектурний патерн MVC.

Model-View-Controller. MVC - це фундаментальний патерн, який знайшов застосування у багатьох технологіях, дав розвиток новим технологіям і щодня полегшує життя розробникам [21].

Вперше патерн MVC з'явився у мові SmallTalk. Розробники мали придумати архітектурне рішення, яке дозволяло б відокремити графічний інтерфейс від бізнес-логіки, а бізнес-логіку від даних. Таким чином, у класичному варіанті MVC складається з трьох частин, які і дали йому назву. Розглянемо їх:

Модель - під моделью, зазвичай розуміється частина містить у собі функціональну бізнес-логіку докладання. Модель має бути повністю незалежною від інших елементів продукту. Модельний шар нічого не повинен знати про елементи дизайну, і яким чином він відобразатиметься. Досягається результат, що дозволяє змінювати представлення даних, як вони відображаються, не чіпаючи саму Модель.

Модель має такі ознаки:

- модель – це бізнес-логіка програми;
- модель має знання про себе саму і не знає про контролерів і уявлення;
- для деяких проектів модель це просто шар даних (DAO, база даних, XML-файл);
- для інших проектів модель — менеджер бази даних, набір об'єктів або просто логіка програми.

Подання (View) - до обов'язків подання входить відображення даних, отриманих від моделі. Однак уявлення не може безпосередньо впливати на модель. Можна говорити, що уявлення має доступом «лише читання» до даних [22].

Подання має такі ознаки: у поданні реалізується відображення даних, що виходять від моделі будь-яким способом; у деяких випадках подання може мати код, який реалізує деяку бізнес-логіку.

Архітектурний патерн MVC зображено на рисунку 2.5.

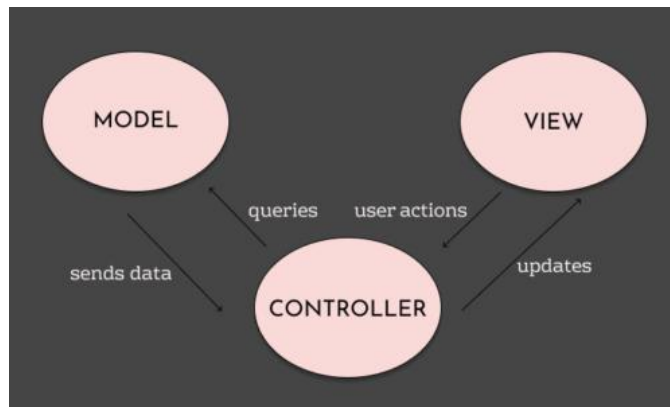


Рисунок 2.5 – Архітектурний патерн MVC

Основна ідея цього патерну в тому, що і контролер і подання залежать від моделі, але модель не залежить від цих двох компонентів.

Ознаки контролера: контролер визначає, які уявлення має бути відображено в даний момент; події подання можуть вплинути тільки на контролер; контролер може вплинути на модель і визначити інше уявлення; можливо кілька уявлень лише одного контролера.

Реалізація: контролер перехоплює подію ззовні та відповідно до закладеної в нього логіки, реагує на цю подію змінюючи модель, за допомогою виклику відповідного методу. Після зміни Модель використовує подію про те, що вона змінилася, і всі підписані на це події Подання, отримавши її, звертаються до Моделі за оновленими даними, після чого їх відображають.

2.5 UML діаграма класів.

Виходячи з поставлених завдань, загальної схеми роботи системи та спираючись на архетектуру MVC, була спроектована UML діаграма класів представлена на рисунку 2.6.

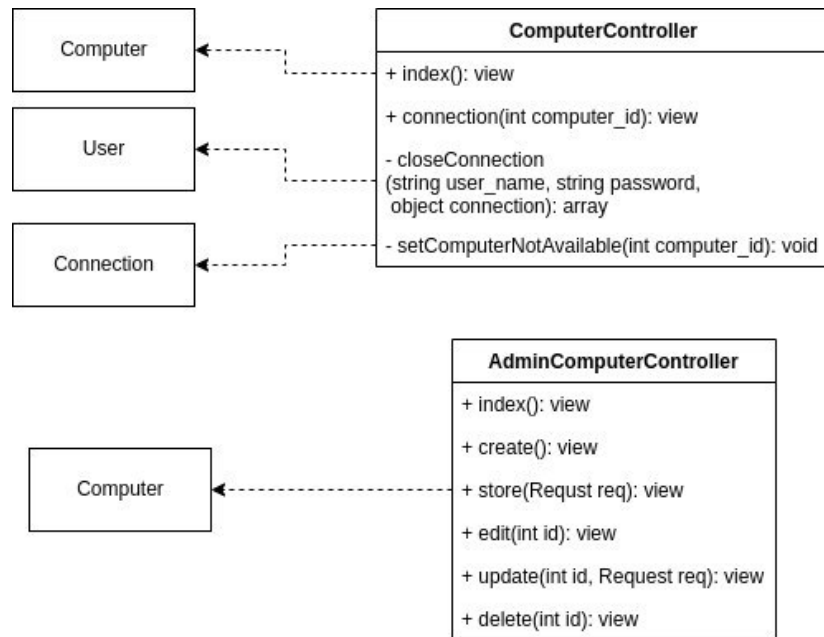


Рисунок 2.6 – UML діаграма класів

Класи **Computer**, **User**, **Connection** – моделі. Вони взаємодіють з базою даних та реалізують базовий функціонал: читання записів із таблиці; додавання запису до таблиці; редагування запису у таблиці; видалення запису з таблиці.

Класи **ComputerController** та **AdminComputerController** є контролерами. Розглянемо їх методи докладніше.

Клас **ComputerController** реалізує взаємодію користувача з комп'ютерами, його методи:

- `index` – повертає подання зі списком всіх доступних для підключення комп'ютерів;
- `connection` – підключає користувача до вибраного комп'ютера та повертає подання з паролем для підключення;
- `closeConnection` – підключається по `ssh` до вибраного комп'ютера та готує його до нового підключення;
- `setComputerNotAvailable` – викликається у разі помилки під час підключення. Змінює статус комп'ютера на недоступний.

Клас `AdminComputerController` реалізує взаємодію адміністратора системи з таблицею комп'ютерів у базі даних. Методи класу `AdminComputerController`:

- `index` – повертає подання зі списком усіх комп'ютерів,
- `create` – повертає подання з формою для створення нового комп'ютера,
- `store` – створює новий комп'ютер,
- `edit` – повертає подання з формою для редагування комп'ютера,
- `update` – оновлює інформацію про комп'ютер,
- `delete` – видаляє комп'ютер.

Виходячи з вищесказаного можна зробити наступний висновок, що функціонал системи умовно можна розділити на дві частини. Перша частина відповідає за взаємодію з користувачем. Друга – дозволяє адміністратору взаємодіяти із системою.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДИСТАНЦІЙНОГО ДОСТУПУ ДО КОМП'ЮТЕРНОЇ НАВЧАЛЬНОЇ ЛАБОРАТОРІЇ

3.1 Середовище програмування

3.1.1 Операційна система Ubuntu

Windows і macOS – дві найпопулярніші операційні системи, доступні сьогодні. Однак Linux з кожним роком набирає все більшої популярності не тільки серед розробників, а й рядових споживачів [23].

Linux - відмінний вибір, якщо ви хочете створити свій віртуальний приватний сервер або спробувати безкоштовну операційну систему. Він поставляється в різних дистрибутивах, таких як Fedora, Debian та CentOS.

Ubuntu є однією з найпопулярніших серед операційні системи, що базуються на ядрі Linux. Ви можете встановити його на персональні комп'ютери та сервери.

На відміну від пропрієтарного програмного забезпечення, Ubuntu має відкритий вихідний код. Це означає, що користувачам дозволено змінювати його код, створювати та встановлювати стільки копій, скільки вони хочуть, і поширювати налаштування будь-де. Вам не потрібно платити за ліцензію на використання Ubuntu.

Ubuntu була представлена у 2004 році британською компанією Canonical. Він був заснований на Debian - популярному на той час дистрибутиві, який було складно встановити. В результаті Ubuntu була запропонована як зручніша альтернатива.

Як менеджер Ubuntu, Canonical відповідає за випуск нової версії Ubuntu кожні шість місяців. Canonical також надає хостинг-сервери для спільноти Ubuntu, що дозволяє людям з усього світу робити свій внесок у тестування програмних помилок, відповідати на запитання та надавати технічну підтримку безкоштовно.

Хоча в Інтернеті доступні різні дистрибутиви Linux, найпопулярнішим серед них є Ubuntu. Це безкоштовна операційна система з відкритим кодом.

Ось шість причин, з яких Ubuntu такий популярний:

- зручність для користувача. — Ubuntu має простий та інтуїтивно зрозумілий інтерфейс.
- надійна безпека — Ubuntu використовує AppArmor та розширені заходи безпеки для запобігання порушенням.
- більше варіантів програмного забезпечення. У Ubuntu можна встановити величезну кількість програм, багато з яких доступні виключно для операційної системи.
- підвищена конфіденційність — Ubuntu забезпечує строгу політику конфіденційності даних та дозволяє користувачам налаштовувати параметри конфіденційності.
- легка продуктивність – інтерфейс Ubuntu за замовчуванням використовує менше 1 ГБ ОЗП. Тому операційна система працює на бюджетних пристроях.
- безкоштовно – Ubuntu – це безкоштовний дистрибутив Linux з відкритим вихідним кодом.

3.1.2 Мова програмування PHP

PHP (англійською HypertextPreprocessor — гіпертекстовий препроцесор, попередня назва: PersonalHomePageTools — засоби для особистої домашньої сторінки) — мова програмування, створена для породження HTML-сторінок зі сторони веб-сервера [24].

PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок разом із Java, .NET, Perl, Python і Ruby.

Веб-сервер інтерпретує код PHP у код HTML, який передає клієнту. Також код PHP можна вбудовувати безпосередньо в код html. На відміну від

мови JavaScript, користувач не бачить PHP-коду, бо браузер клієнта отримує готовий html-код.

Давайте проаналізуємо, що відбувається, коли користувач звертається до сервера. Він пише в рядку адреси браузера адресу сторінки чи переходить за якимось посиланням, тобто створює запит. Що робить сервер?

У найпростішому випадку викликають HTML сторінку — текстовий файл. Браузер, посилає запит на сервер, отримує код сторінки, додатково завантажує необхідні файли, прописані в коді сторінки і необхідні для відображення її дизайну та її коректної роботи, та показує результат у своєму вікні. Такі сторінки мають статичний характер (дивіться рисунок 3.1).



Рисунок 3.1 – Виклик HTML сторінки

Тепер розглянемо випадок, коли викликають сторінку, написану мовою PHP. У цьому випадку відбувається зовсім інше. При зверненні до сервера Web машина обробляє дані, отримані з файлу *.php, після чого сервер відправляє результат опрацювання цих даних у браузер користувача. Такі сторінки називають динамічними (дивіться рисунок 3.2).

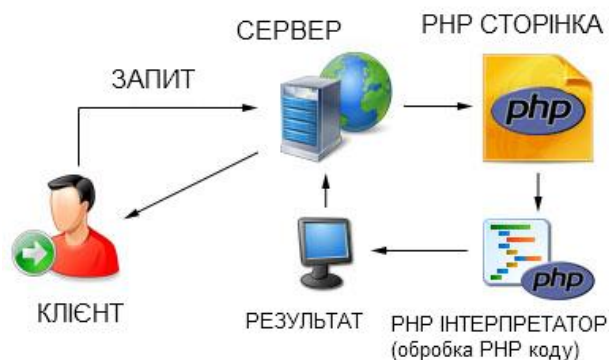


Рисунок 3.2 – Виклик PHP сторінки

Основні області застосування PHP коду. Написання скриптів для виконання на стороні сервера. PHP традиційно і найширше використовують саме таким чином. Потрібно запуснути веб-сервер з встановленою на ньому мовою PHP. Через веб-сервер можна пропускати сторінку з кодом PHP, та проглядати результат її роботи через веб-браузер. Все це можна робити навіть на домашньому комп'ютері. Хоча існує можливість опробувати код PHP, використовуючи online сервіс. Наприклад, Coding Ground.

PHP можна використовувати при більшості операційних систем: Linux, багато варіантів Unix (включаючи HP-UX, Solaris та OpenBSD), Microsoft Windows, Mac OS X, RISC OS та, можливо, інших.

PHP має також підтримку для більшості сучасних веб-серверів: Apache, IIS та багатьох інших. Окрім цього, підтримка є на будь-якому веб-сервері, що використовує бінарні коди FastCGI — клієнт-серверного протоколу взаємодії веб-сервера та програм.

Одна з найсильніших та найістотніших особливостей PHP — це підтримка широкого кола баз даних: Adabas D, dBase, Direct MS-SQL, Empress, FilePro (read-only), FrontBase, Hyperwave, IBM DB2, Informix, Ingres, InterBase, mSQL, MySQL, ODBC, Oracle (OCI7 и OCI8), Ovrimos, PostgreSQL, Solid, Sybase, Unix dbm.

3.1.3 База даних MySQL

MySQL — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення

динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [25].

3.1.4 Web-сервер Nginx

Nginx, вляє собою веб-сервер з відкритим вихідним кодом, який з моменту свого початкового успіху як веб-сервер тепер також використовується як зворотний проксі-сервер, кеш HTTP і балансувальник навантаження [26].

Деякі відомі компанії, що використовують Nginx, включають Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple . , Intel та багато інших.

Спочатку Nginx було створено Ігорем Сисоєвим, а його перший публічний випуск відбувся у жовтні 2004 року. Спочатку Ігор задумав програмне забезпечення як вирішення проблеми C10k, яка пов'язана з проблемою продуктивності при обробці 10 000 одночасних підключень.

Оскільки його коріння лежить в оптимізації продуктивності при масштабуванні, Nginx часто перевершує інші популярні веб-сервери в тестах продуктивності, особливо в ситуаціях зі статичним контентом та/або великою кількістю одночасних запитів.

Nginx створений для забезпечення низького використання пам'яті та високого паралелізму. Замість створення нових процесів для кожного веб-запиту, Nginx використовує асинхронний, керований подіями підхід, при якому запити обробляються в одному потоці.

У Nginx один головний процес може керувати кількома робочими процесами. Майстер підтримує робочі процеси, тоді як робітники виконують фактичну обробку. Оскільки Nginx є асинхронним, кожен запит може виконуватися одночасно виконавцем, не блокуючи інші запити.

Деякі загальні функції, що спостерігаються в Nginx, включають:

- зворотний проксі з кешуванням;

- IPv6;
- балансування навантаження;
- підтримка FastCGI з кешуванням;
- веб-сокети;
- обробка статичних файлів, індексних файлів та автоіндексація;
- TLS/SSL із SNI.

3.1.5 Laravel фреймворк

Laravel - це кроссплатформенний PHP-фреймворк для створення веб-додатків[27].

Але справжнє розуміння переваг Laravel та його використання потребує глибшого занурення. Laravel дозволяє розробнику скористатися великою бібліотекою попередньо запрограмованих функцій (таких як автентифікація, маршрутизація та HTML-шаблони). Доступ до цієї бібліотеки спрощує швидке створення надійних веб-застосунків і зводить до мінімуму обсяг необхідного коду.

Laravel пропонує високофункціональне середовище розробки, а також інтуїтивно зрозумілий та виразний інтерфейс командного рядка. Крім того, Laravel використовує об'єктно-реляційне зіставлення (ORM) для більш простого доступу до даних та маніпулювання ними.

Програми Laravel добре масштабуються та мають просту в обслуговуванні кодову базу. Розробники також можуть легко додавати функціональні можливості у свої програми завдяки модульній системі упаковки Laravel та надійному керуванню залежностями.

Laravel - це PHP-фреймворк, що використовує мову сценаріїв, а не строго мову програмування PHP. Хоча мови сценаріїв та мови програмування пов'язані між собою, вони мають кілька помітних відмінностей, насамперед у простоті використання та швидкості виконання.

Особливості фреймворку Laravel, які виділяють його серед інших.

1. Підтримка MVC та об'єктно-орієнтований підхід. Перша і найкраща перевага використання фреймворку Laravel полягає в тому, що він дотримується архітектурного шаблону на основі моделі, представлення та контролера і має виразний красивий синтаксис, який робить його об'єктно-орієнтованим.

2. Вбудована автентифікація та авторизація. Laravel надає готову конфігурацію для системи автентифікації та авторизації. Тобто всього за кілька ручних команд ваш додаток буде оснащено безпечною автентифікацією та авторизацією.

3. Система пакування. Система упаковки має справу з кількома допоміжними програмами або бібліотеками, які допомагають веб-програмі автоматизувати процес. Laravel використовує композитор як менеджер залежностей, який керує всією інформацією, необхідною для управління пакетами. Пакети – чудовий спосіб прискорити розробку, оскільки вони надають необхідну нам функціональність «з коробки». Image, панель налагодження Laravel та помічник Laravel IDE – одні з найкращих пакетів Laravel.

4. Консоль ремісника. Laravel має власний інтерфейс командного рядка, який називається Artisan. Зазвичай Artisan використовується для публікації ресурсів пакетів, управління міграціями баз даних, заповнення та створення стандартного коду для нових контролерів, моделей та міграцій. Ця функція звільняє розробника від створення належних кістяків коду. Можна розширити функціональність і можливості Artisan, впровадивши нові команди користувача.

5. Промовистий ORM. Eloquent ORM - це вбудована реалізація ORM в Laravel. Laravel має найкращий об'єктно-реляційний Mapper у порівнянні з іншими фреймворками. Це об'єктно-реляційне зіставлення дозволяє взаємодіяти з об'єктами вашої бази даних і відносинами бази даних, використовуючи виразний синтаксис.

б. Механізм шаблонів. Laravel поставляється із вбудованим механізмом шаблонів, відомим як Blade Template Engine. Механізм шаблонів Blade поєднує один або кілька шаблонів з моделлю даних для створення результуючих уявлень, роблячи це шляхом перетворення шаблонів в кешований PHP код для підвищення продуктивності. Blade також надає набір власних керуючих структур, таких як умовні оператори та цикли, які внутрішньо зіставляються зі своїми аналогами PHP.

Крім цих функцій, Laravel також має офіційні пакети, які стануть у нагоді при інтеграції різних функцій у додаток.

3.1.6 Bootstrap

Bootstrap - це відкритий і безкоштовний HTML, CSS і JS фреймворк, який використовується веб-розробниками для швидкої верстки адаптивних дизайнів сайтів та веб-додатків [28].

Фреймворк Bootstrap використовується по всьому світу не тільки незалежними розробниками, але іноді й цілими компаніями. На Bootstrap створено дуже багато різних сайтів, подивитися їх можна на сторінці Bootstrap Expo.

Основна область його застосування - це фронтенд розробка сайтів та інтерфейсів адмінок. Серед аналогічних систем (Foundation, UIKit, Semantic UI, InK і ін.) Фреймворк Bootstrap є найпопулярнішим .

Чому Bootstrap такий популярний? Це пов'язано з тим, що він дозволяє верстати сайти в кілька разів швидше , ніж на «чистому» CSS і JavaScript. А в нашому світі, час - це дуже цінний ресурс. Ще один його аспект - доступність . Вона зводиться до того, що надає можливість навіть початківцю веб-розробнику (без глибоких знань і достатньої практики) створювати досить якісні макети .

Фреймворк Bootstrap - це набір CSS і JavaScript файлів. Щоб його використовувати ці файли необхідно просто підключити до сторінки. Після

цього вам стануть доступні інструменти даного фреймворка: колонкова система (сітка Bootstrap), класи і компоненти.

3.1.7 Docker та Docker Compose

Docker – це інструмент з відкритою платформою, що спрощує створення, розгортання та виконання програм за допомогою контейнерів. Контейнери дозволяють нам відокремити програми від інфраструктури, щоб ми могли швидше розгортати програми/програмне забезпечення[29].

Ми можемо керувати нашою інфраструктурою так само, як ми керуємо нашими програмами. Docker нагадує віртуальну машину, але створює нову цілу віртуальну машину; це дозволяє нам використовувати одне і те ж ядро Linux.

Перевага платформи Docker полягає у швидшому відправленні, тестуванні та розгортанні коду, що дозволяє ефективно скоротити час між написанням коду та його виконанням у робочому середовищі.

І головне, що важливо в Docker, це те, що це відкритий вихідний код, тобто будь-який може використовувати його і може зробити свій внесок у Docker, щоб зробити його простіше і додати більше функцій, яких у ньому немає.

Докер Платформа. Перевага докера полягає в тому, щоб зібрати пакет і запустити додаток у середовищі пісочниці, згаданій у контейнері.

Система контейнерів докерів використовує віртуалізацію операційної системи для використання та поєднання компонентів системи додатків, які підтримують кожну стандартну машину Linux.

Фактори ізоляції та безпеки дозволяють нам виконувати безліч контейнерів паралельно у даній системі.

Контейнери мають невеликий розмір, оскільки їм не потрібні додаткові ресурси HyperV або VMware, вони працюють безпосередньо в ядрі машини. Ми навіть можемо запускати контейнери Docker на машинах, які є віртуальними/гіпермашинами.

Основні компоненти Docker. Ядро Docker складається з Docker Engine, контейнерів Docker, образів Docker, клієнта Docker, демона Docker та ін.

Docker Compose — це інструмент, який використовується для визначення та запуску кількох контейнерів у програмах Docker. Docker composes використовується для створення файлу компонування для налаштування служб додатків. Після цього однією командою створюємо та запускаємо всі сервіси з нашої конфігурації.

Docker Compose – дуже корисний інструмент для середовищ розробки, тестування та підготовки.

Docker Compose – це триетапний процес.

- визначте середовище програми за допомогою файлу Dockerfile, щоб його можна було відтворити у будь-який час та в будь-якому місці.
- визначте служби у docker-compose.yml, після чого ми зможемо працювати разом у ізольованому середовищі.
- після цього за допомогою docker-compose up і Compose запуститься програма.
- сервер, який є постійно працюючою службою, що називається процесом демона.
- REST API, який взаємодіє з програмами, щоб спілкуватися з демоном та давати йому вказівки, що робити.

3.2 Розгортання системи

У розділі розглянемо розгортання системи на комп'ютерах навчальної лабораторії. Система розроблена під операційну систему Windows 10. Для успішного розгортання системи необхідно виконати на кожному комп'ютері такі дії.

1. Встановити RustDesk.
2. Встановити файли, що виконуються.
3. Встановити сервер OpenSSH.

4. Створити завдання у Windows.

Розглянемо кожен пункт детальніше.

3.2.1 Встановлення RustDesk.

Встановлення RustDesk мало чим відрізняється від встановлення будь-якого іншого програмного забезпечення. Заходимо на офіційний сайт rustdesk.com та натискаємо кнопку скачати «Download». На рисунку 3.3 зображено офіційний сайт RustDesk.

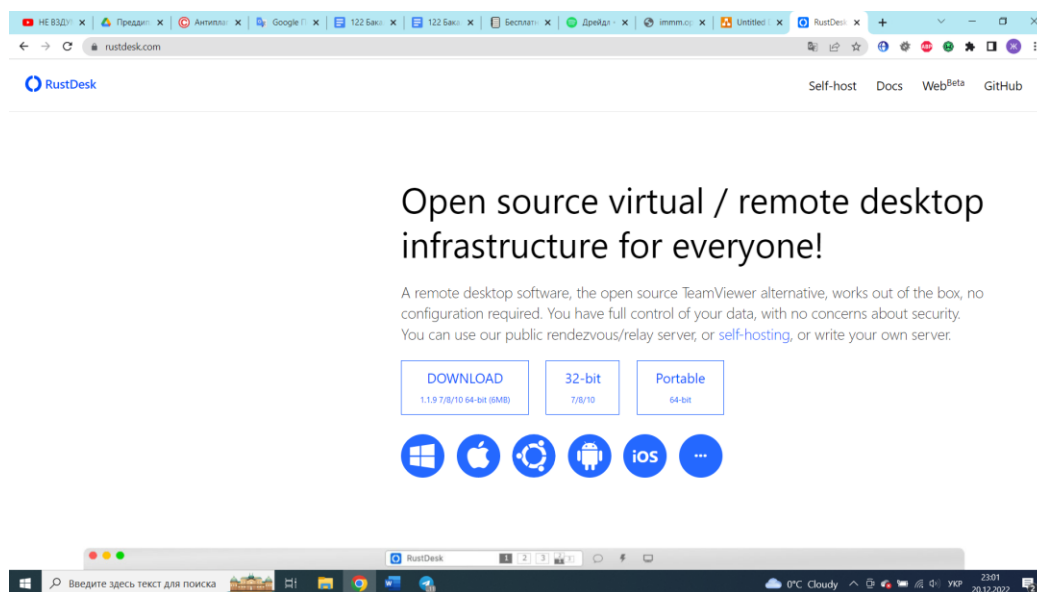


Рисунок 3.3 – Офіційний сайт RustDesk

Завантажується архів zip. Заходимо до нього та запускаємо установник. На рисунку 3.4 зображено запуск установника RustDesk.

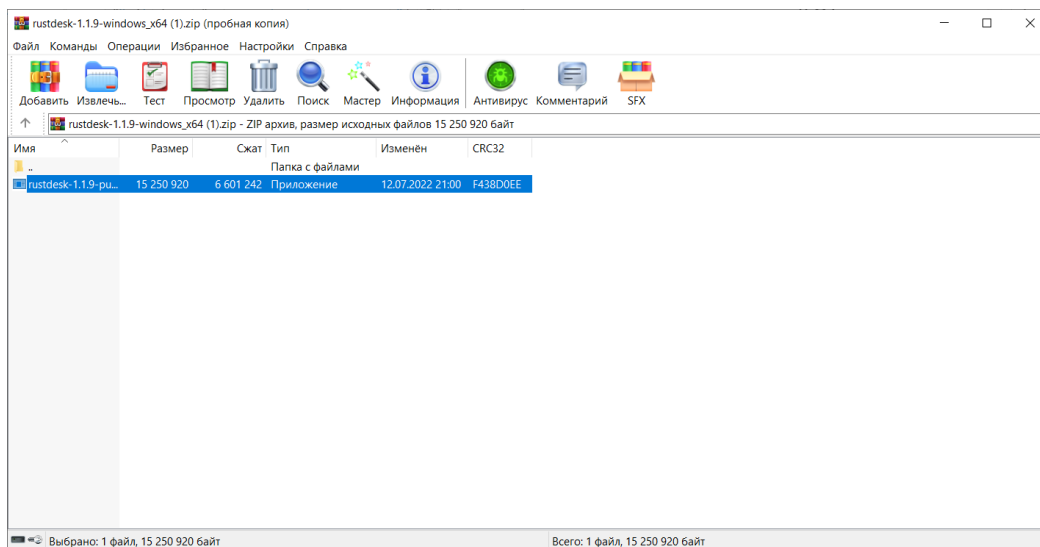


Рисунок 3.4 – Запуск установника RustDesk

У установнику вибираємо папку та за потребою додаткові опції. І натискаємо кнопку «Прийняти та встановити». На рисунку 3.5 зображено запуск установник RustDesk.

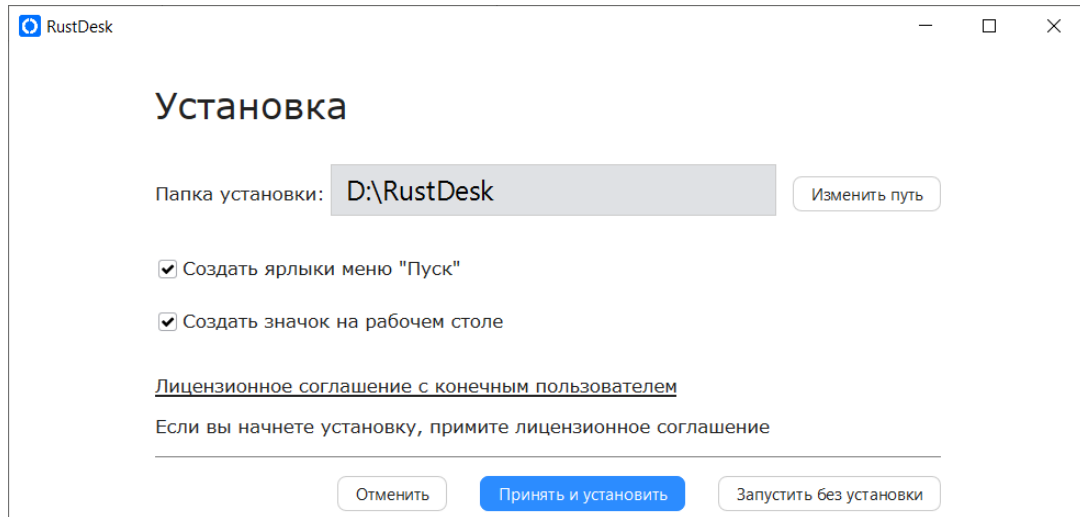


Рисунок 3.5 – Установник RustDesk

Після завершення встановлення ми побачимо наступне вікно. На рисунку 3.6 зображено запуск інтерфейс RustDesk.

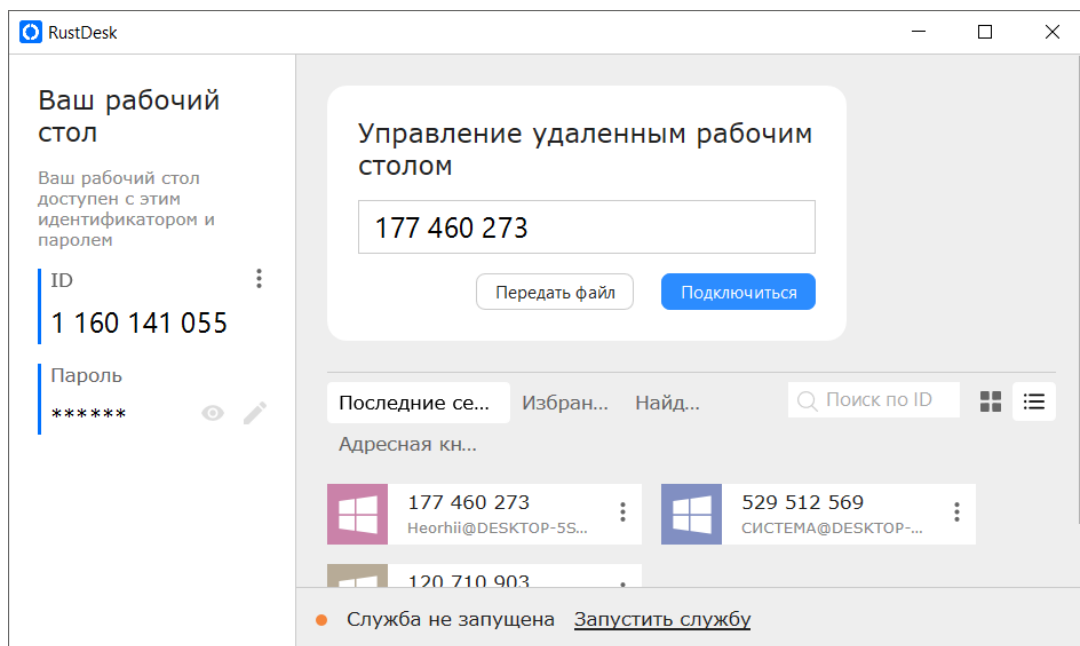


Рисунок 3.6 – Интерфейс RustDesk

На цьому встановлення можна вважати завершеним. Програма готова до роботи.

3.2.2 Встановлення та налаштування пакета виконуваних файлів

Виконувані файли написані мовою Python. Вони необхідні для того, щоб сервер, запускаючи їх віддалено, міг взаємодіяти з програмою RustDesk. Пакет включає три файли: `kill_RustDesk.py`, `change_pass.py`, `startRustDesk.py`. Їх необхідно встановити у кореневий каталог користувача. На рисунку 3.7 зображено кореневий каталог користувача.

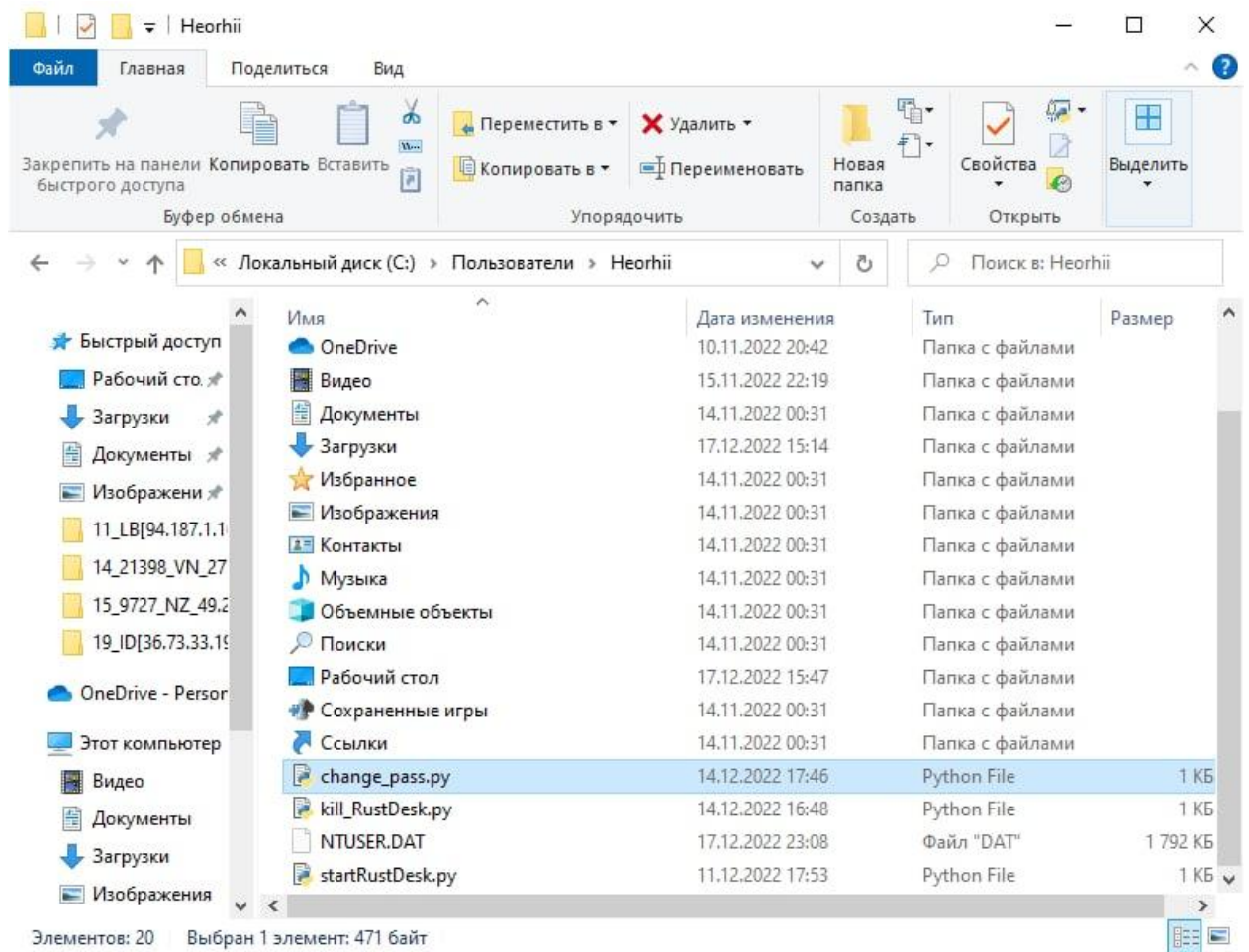


Рисунок 3.7 – Кореневий каталог користувача

Далі потрібно редагувати файл `startRustDesk.py`. Відкриваємо його у блокноті. Він має такий зміст.

```
import os
def run_rust_desk():
    os.startfile('D:\RustDesk\RustDesk.exe')
run_rust_desk()
```

Змінюємо рядок `D:\RustDesk\RustDesk.exe` у відповідність з нашим шляхом до програми RustDesk. На рисунку 3.8 зображено шлях до програми RustDesk.

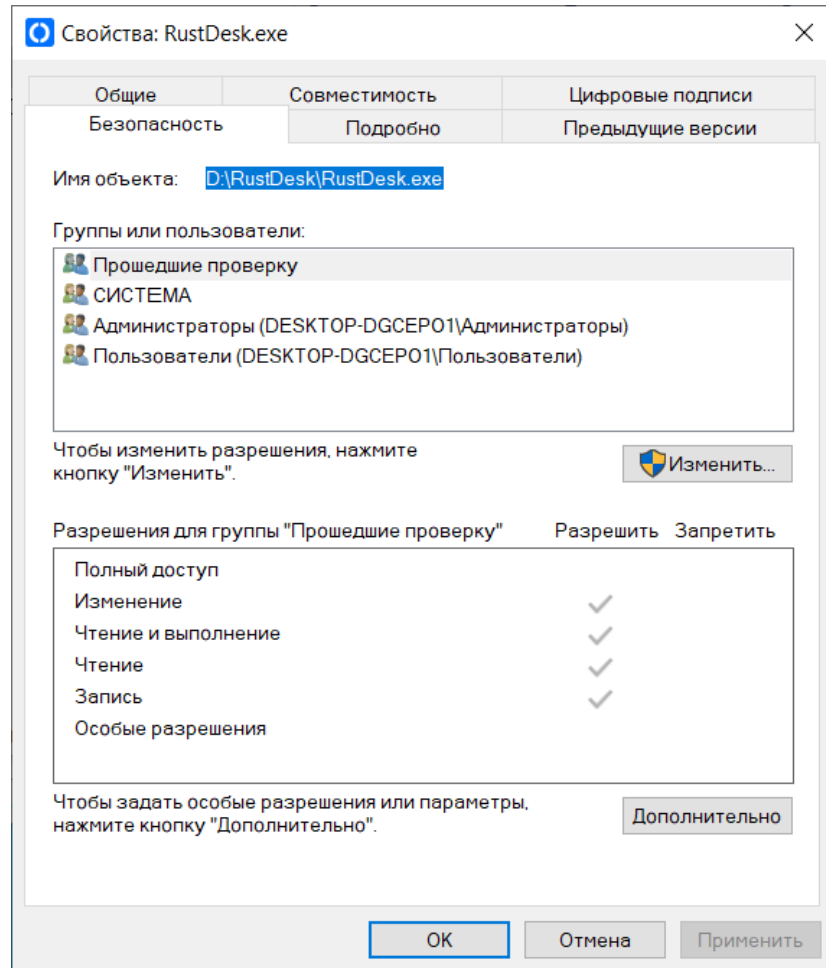


Рисунок 3.8 – Шлях до програми RustDesk

На цьому встановлення та налаштування пакета виконуваних файлів завершено.

3.2.3 Встановлення сервера OpenSSH

Сервера OpenSSH забезпечує можливість підключення до цього комп'ютера за протоколом SSH. Для його встановлення переходимо до Пуск > Параметри Приложения > Дополнительные компоненты > Добавить компонент. І вибираємо сервер OpenSSH. На рисунку 3.9 зображено встановлення сервера OpenSSH.

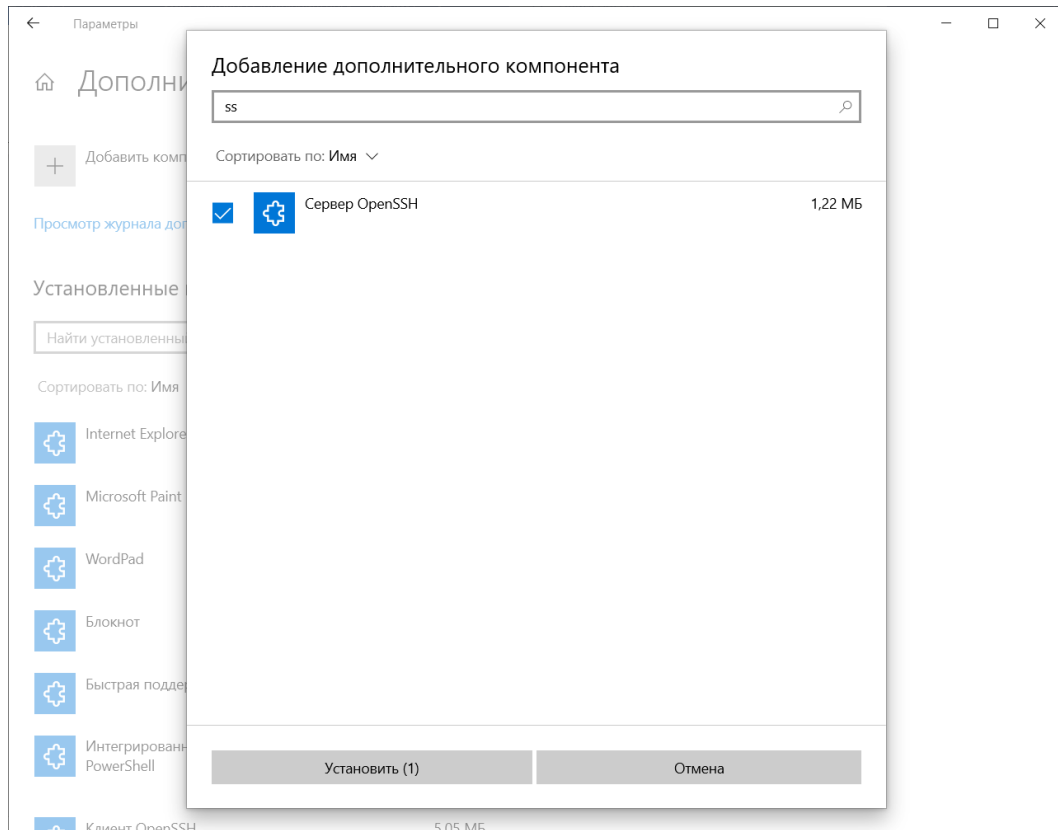


Рисунок 3.9 – Встановлення сервера OpenSSH

Далі потрібно змінити тип запуску служби `sshd` на автоматичний та запустити службу за допомогою PowerShell:

```
Set-Service -Name sshd -StartupType 'Automatic'
Start-Service sshd
```

За допомогою `netstat` переконайтеся, що тепер в системі запущено SSH сервер і чекає на підключення на порту TCP:22 :

```
netstat -na | find ":22"
```

Перевірте, чи увімкнено правило брандмауера (Windows Defender Firewall), яке дозволяє вхідні підключення до Windows через порт TCP/22.

```
Get-NetFirewallRule -Name *OpenSSH-Server* |select Name, DisplayName, Description, Enabled
```

Якщо правило вимкнено (стан `Enabled=False`) або відсутній, ви можете створити нове вхідне правило командою `New-NetFirewallRule`:

```
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
```

На цьому встановлення та налаштування сервера OpenSSH завершено.

3.2.4 Створення задач в Windows

Стандартними засобами Windows створюємо задачу з ім'ям startRustDesk. На рисунку 3.10 зображено задача startRustDesk (Загальні).

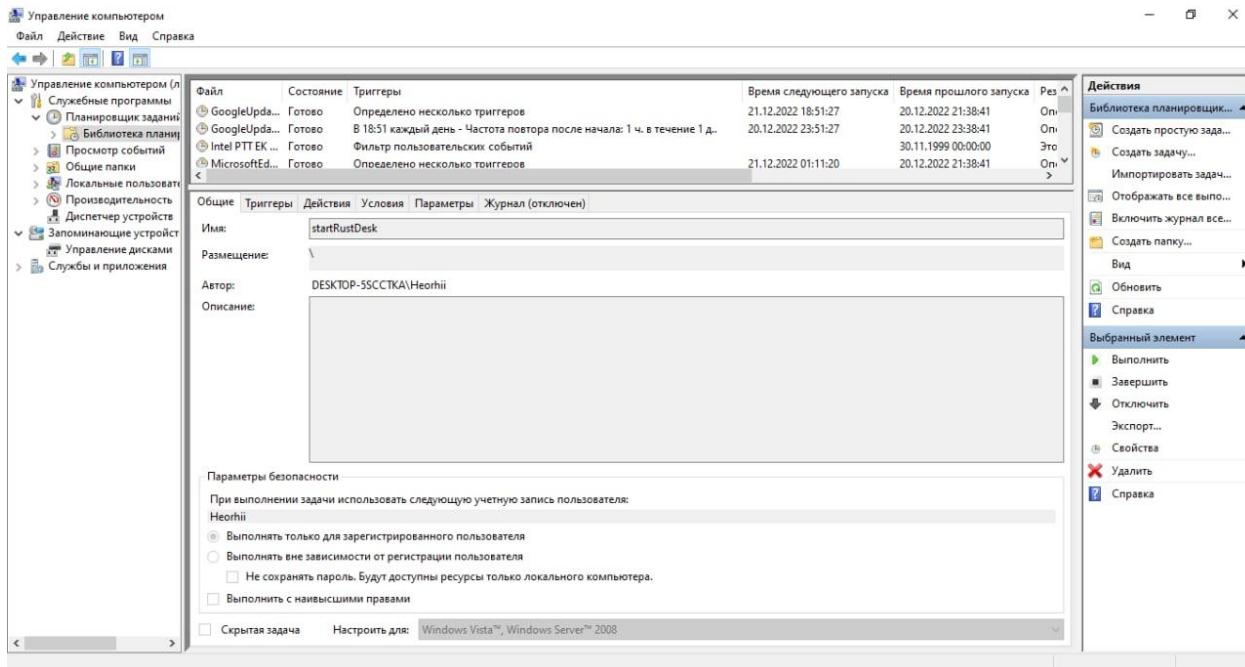


Рисунок 3.10 – Задача startRustDesk (Загальні).

Як дія вибираємо запуск скрипта startRustDesk.py. Для цього треба вказати шлях до нього. На рисунку 3.11 зображено задача startRustDesk (Дія).

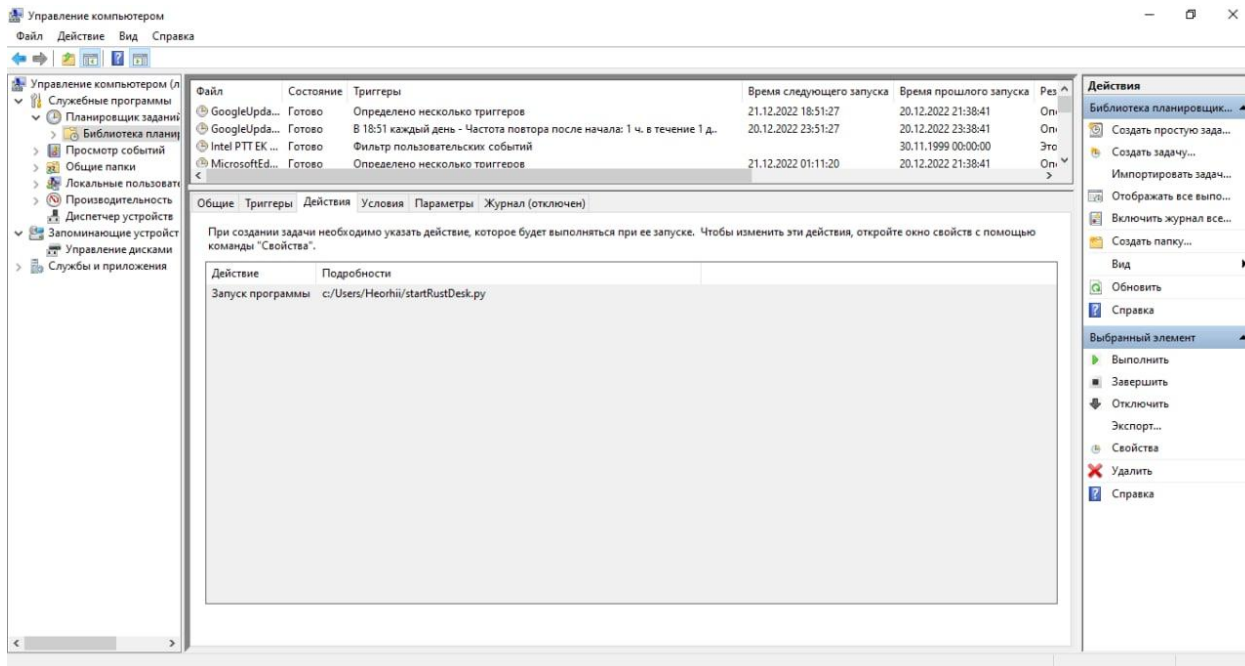


Рисунок 3.11 – задача startRustDesk (Действие).

На цьому створення задачі завершено.

Виконавши ці чотири пункти на комп'ютері навчальної лабораторії, ми налаштували його для використання в розробленій системі віддаленого доступу.

3.3 Моделювання роботи системи

3.3.1 Панель адміністратора

Панель адміністратора має вигляд як представлено рис. 3.12. У ній адміністратор системи може додавати, редагувати та видаляти комп'ютери.

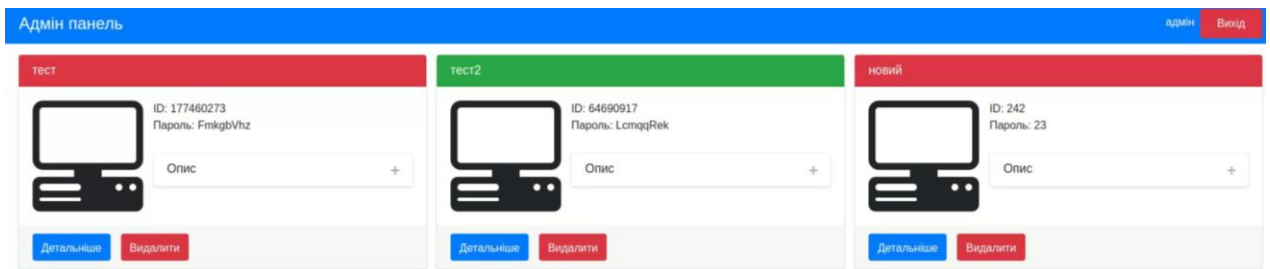


Рисунок 3.12 – Панель адміністратора.

Створимо новий комп'ютер з даними, представленими на малюнку 3.13.

 The image shows a form titled 'Вікно редагування комп'ютера' (Computer editing window). The form contains several input fields: 'Ім'я' (Name) with the value 'test'; 'IP-адреса комп'ютера' (Computer IP address) with the value '192.168.0.107'; 'ID у програмі RustDesk' (RustDesk program ID) with the value '177460273'; 'Ім'я облікового запису в операційній системі' (OS username) with the value 'Neorhii'; 'Пароль до облікового запису в операційній системі' (OS password) with the value '1'; and 'Опис' (Description) with the text '"But I must explain to you how all this mistaken idea human happiness. No one rejects, dislikes, or avoids'. There is a checked checkbox labeled 'Доступний' (Available). At the bottom of the form, there are two buttons: 'Зберегти' (Save) in green and 'скасування' (Cancel) in red.

Рисунок 3.13 – Вікно редагування комп'ютера.

Тепер комп'ютер з ір-адресою 192.168.0.107 доступний для підключення студентів.

3.3.2 Реєстрація та аутентифікація в системі

Для реєстрації необхідно надати електронну пошту та придумати пароль. Для користування системою також необхідно підтвердити адресу електронної пошти. Зареєструємось у системі. На рисунку 3.14 зображено вікно реєстрації.

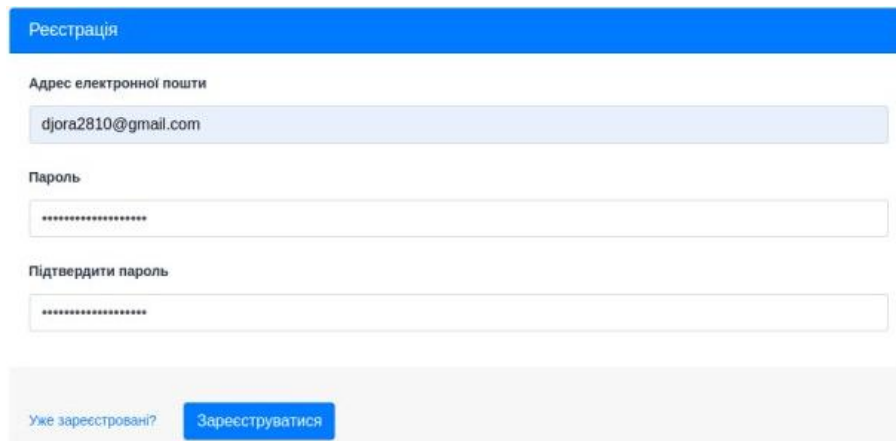


Рисунок 3.14 – Вікно реєстрація

Після реєстрації ми побачимо повідомлення про необхідність підтвердити адресу електронної пошти. На рисунку 3.15 зображено Сповідження.

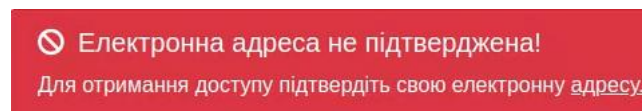


Рисунок 3.15 – Сповідження.

Після підтвердження адреси це повідомлення зникне і ми зможемо скористатися системою.

3.3.3 Підключення до комп'ютера віддаленої навчальної лабораторії

Завершивши реєстрацію, користувач побачить інтерфейс як представлено на малюнку 3.16.

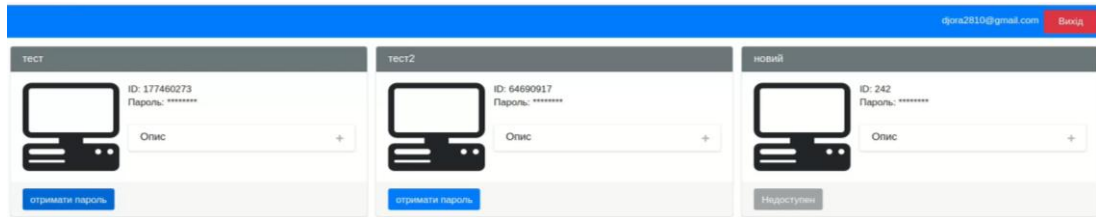


Рисунок 316 – Інтерфей користувача.

Користувач може вибрати комп'ютер та підключитися до нього. Підключимося до комп'ютера з іменем «test» натиснувши на кнопку «Отримати пароль». Після чого система надасть нам ID та пароль для підключення, як представлено на малюнку 3.17.

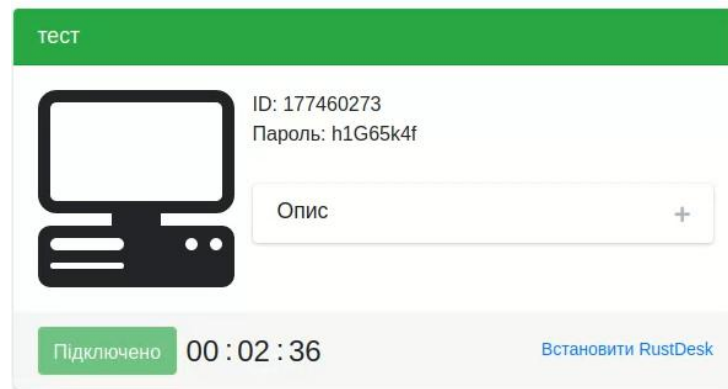


Рисунок 3.17 – Отримання пароля.

Далі заходимо в RustDesk і підключаємося за наданим ID, в нашому випадку це 177460273. RustDesk запросить пароль, вводим раніше отриманий пароль, в нашому випадку це h1G65k4f.

Маючи віддалений доступ до комп'ютера, заходимо в консоль і вводим команду `ipconfig`, щоб перевірити ip-адресу і переконатися в тому, що підключилися до потрібного комп'ютера. На рисунку 3.18 зображено вікно віддаленого доступу.

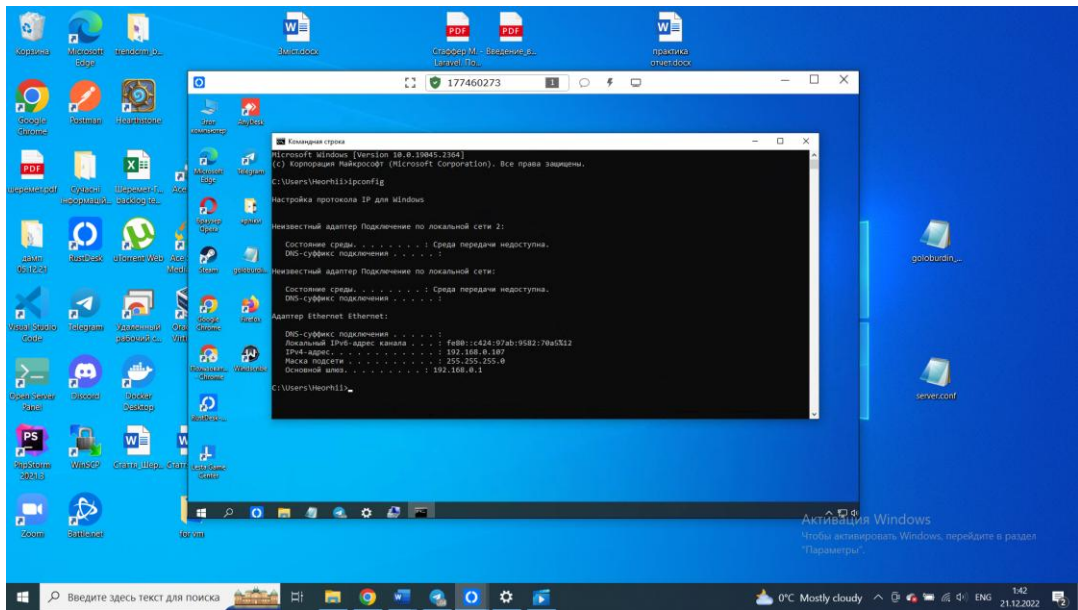


Рисунок 3.18 – Вікно віддаленого доступу.

Як видно на малюнку ір-адреса 192.168.0.107. Такий, як ми вказували при створенні комп'ютера в панелі адміністратора. Відтак система працює коректно та готова до використання.

3.3.4 Черга на підключення

Також у системі існує можливість зайняти чергу у разі коли потрібний нам комп'ютер зайнятий іншим користувачем. Зайнятий комп'ютер виглядає як на малюнку 3.19.

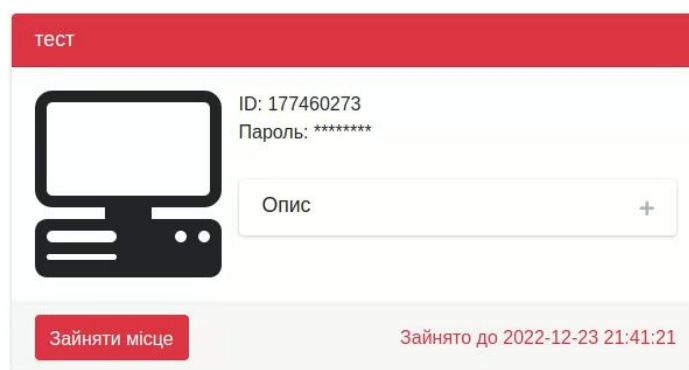


Рисунок 3.19 – Зайнятий комп'ютер.

При натисканні на кнопку "Зайняти місце" ми зайемо місце в черзі. Перебування у черзі виглядає як представлено малюнку 3.20.

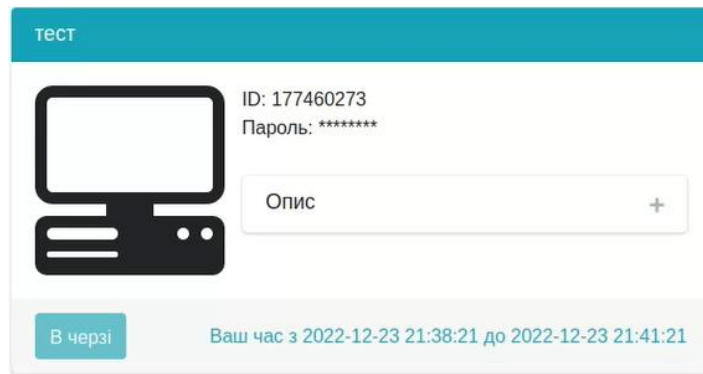


Рисунок 3.20 – Знаходження у черзі.

При настанні нашої черги система надасть нам пароль і ми зможемо отримати віддалений доступ як описано в пункті 3.3.2.

ВИСНОВКИ

В межах кваліфікаційної роботи виконана мета роботи – організовано та автоматизовано дистанційний доступ до комп'ютерної навчальної лабораторії за допомогою web-технологій.

При досягненні поставленої мети в роботі розв'язані наступні задачі:

- розроблено систему реєстрації, автентифікації та авторизацію користувачів;
- надана можливість віддаленого керування курсором та клавіатурою;
- надана можливість віддаленого запуску програмного забезпечення;
- надана можливість обміну файлами між персональним комп'ютером користувача та комп'ютером навчальної лабораторії;
- забезпечено почергове використання комп'ютерів студентами.

Проведено аналіз існуючих програм дистанційного доступу до комп'ютерних мереж. Серед них була обрана і взята за основу програма, що найбільш підходить за функціоналом і вимогами. Спроектована загальна схема роботи системи. Побудована база даних.

Розроблено програмний додаток який:

- забезпечує реєстрацію, автентифікацію та авторизацію користувачів у системі;
- надає можливість віддаленого керування курсором та клавіатурою;
- надає можливість віддаленого запуску програмного забезпечення;
- надає можливість обміну файлами між персональним комп'ютером користувача та комп'ютером навчальної лабораторії;
- забезпечує почергове використання комп'ютерів студентами.

Також проведено моделювання роботи системи дистанційного доступу до комп'ютерної навчальної лабораторії. Моделювання продемонструвало основні функції системи та підтвердило її працездатність.

Отримані в даній роботі результати можуть бути корисні освіти а й для дистанційного бізнес – коворкінга у невеликих фірмах, наукових установах. Результати роботи можуть служити основою для подальших розробок у цій галузі.

Перелік посилань

1. Шеремет Г.О. Організація дистанційного доступу до комп'ютерної навчальної лабораторії за допомогою веб-технологій. *Інформатика та математичні методи моделювання*. 2022. №3-4
2. Haucke T.A.N., Östmarck A. An Analysis of the Co-working Space Industry in Stockholm from an Entrepreneurial Perspective. URL: <https://kth.diva-portal.org/smash/get/diva2:1190270/FULLTEXT01.pdf>
3. Fahrizal A.J.C., Juan B.M., Ramdhani R., Hadiwiroso S., Hamdi E., Indradewa R., Abadi F. Strategic Formulation Analysis of Coworking Space Businesses Using Containers. URL: https://www.ijrrjournal.com/IJRR_Vol.9_Issue.3_March2022/IJRR022.pdf
4. Endrissat N., Vandelannoitte A.L. From sites to vibes: Technology and the spatial production of coworking spaces. URL: <https://hal.archives-ouvertes.fr/hal-03332209/document>
5. Kraus S., Bouncken R.B., Görmar L., González-Serrano M.H., Calabuig F. Coworking spaces and makerspaces: Mapping the state of research. URL: <https://www.econstor.eu/bitstream/10419/260976/1/1796986151.pdf>
6. Hofeditz L., Mirbabaie M., Stieglitz S. Virtually Extended Coworking Spaces? The Reinforcement of Social Proximity, Motivation and Knowledge Sharing Through ICT. URL: <https://arxiv.org/ftp/arxiv/papers/2012/2012.09538.pdf>
7. Roche M., Oettl A., Catalina C. (Co-) Working in Close Proximity: Knowledge Spillovers and Social Interactions. URL: [https://www.hbs.edu/ris/Publication %20Files/21-024rev2-11-22_4cf1fb54-e60b-41e6-8611-985031c999ba.pdf](https://www.hbs.edu/ris/Publication%20Files/21-024rev2-11-22_4cf1fb54-e60b-41e6-8611-985031c999ba.pdf)
8. Challenges of Distance Learning for Students. URL: <https://www.nu.edu/blog/challenges-of-distance-learning-for-students>

9. Удаленная работа на Microsoft Azure. URL: <https://softline.kg/solutions/microsoft-solutions/udalennaya-rabota-na-microsoft-azure>
10. Always On VPN deployment infrastructure. URL: <https://learn.microsoft.com/en-us/windows-server/remote/remote-access/vpn/always-on-vpn/always-on-vpn-technology-overview>
11. Azure Virtual Desktop. URL: <https://azure.microsoft.com/en-us/products/virtual-desktop/>
12. Azure Active Directory Application Proxy documentation. URL: <https://learn.microsoft.com/en-us/azure/active-directory/app-proxy/>
13. Welcome to Remote Desktop Services. URL: <https://learn.microsoft.com/en-us/windows-server/remote/remote-desktop-services/welcome-to-rds>
14. Trustviewer. URL: <http://trustviewer.com>
15. LiteManager. URL: <http://litemanager.ru>
16. Remote Utilitie. URL: <https://www.remoteutilities.com>
17. Как получить доступ к другому компьютеру, используя Удаленный рабочий стол Chrome. URL: <https://support.google.com/chrome/answer/1649523>
18. TeamViewer. URL: <https://www.teamviewer.com>
19. UltraVNC. URL: <https://uvnc.com>
20. RustDesk. URL: <https://rustdesk.com>
21. Паттерны для новичков: MVC vs MVP vs MVVM. URL: <https://habr.com/ru/post/215605/>
22. How Laravel implements MVC and how to use it effectively. URL: <https://blog.pusher.com/laravel-mvc-use/>
23. What Is Ubuntu? A Quick Beginner's Guide. URL: <https://www.hostinger.com/tutorials/what-is-ubuntu>
24. Романов В. Мова програмування PHP, середовище програмування. URL: http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html

25. Справочное руководство по MySQL. URL:
http://www.mysql.ru/docs/man/Database_use.html
26. What Is Nginx? A Basic Look at What It Is and How It Works. URL:
<https://kinsta.com/knowledgebase/what-is-nginx/>
27. The Laravel PHP Framework – Web App Construction for Everyone. URL:
<https://kinsta.com/knowledgebase/what-is-laravel/>
28. 10 Reasons Why Laravel Is The Best PHP Framework For 2022. URL:
<https://www.clariontech.com/blog/10-reasons-why-laravel-is-the-best-php-framework-for-2019>
29. Docker Overview – A Complete Guide URL: <https://hackernoon.com/docker-overview-a-complete-guide-43decd218eca>