

## Практична робота №1

Мета: Створити документ формату pdf

Завдання В цьому документі написати кілька речень з фактами про себе (обов'язково написати прізвище і групу). Наприклад: Я студент групи 124-22-1 Микола Іванов, і я маю кота Снікєрса. А ще я обожнюю баскетбол.

Я студент групи **122-22-1 Андрушак Кирило Олегович**. Люблю спорт, особливо футбол і теніс. Також цікавлюсь програмуванням та сучасними гаджетами. У вільний час люблю слухати музику і малювати.

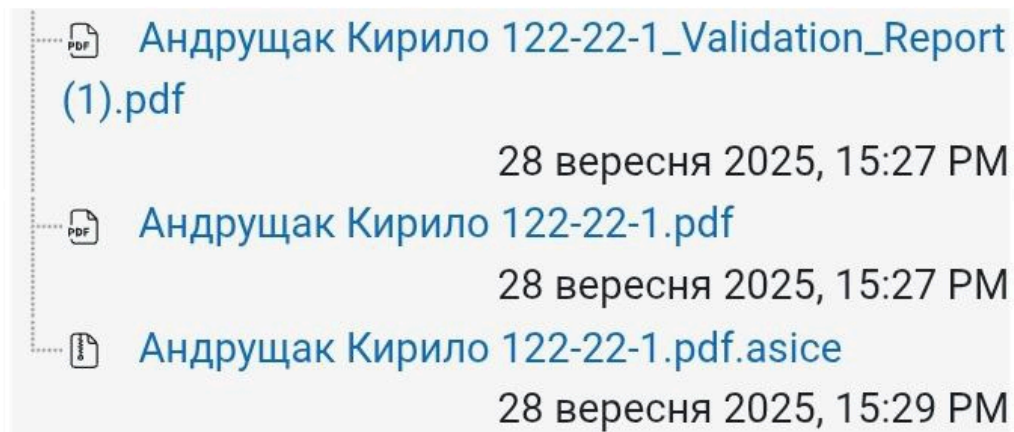


Рис. 1.1-1.2 Зміст пдф файлу та результат підписання

## Практична робота №2

Мета: Створити обліковий запис (ваше прізвище англ мовою) на сайті <https://github.com/>

Завдання: Додати папку APZ (доступ - public) яка буде містити всі практичні по курсу "Аналіз програмного забезпечення"

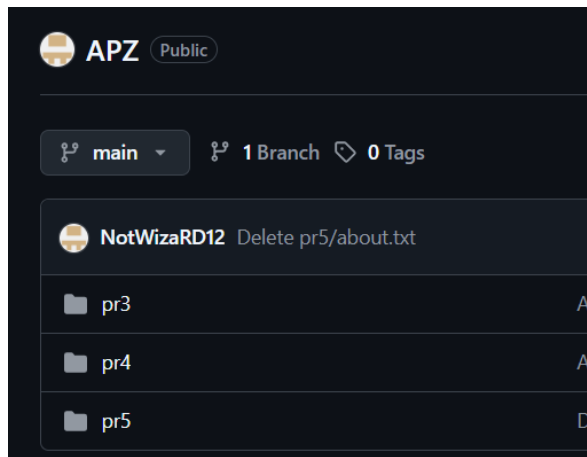


Рис.1 - Результат створення облікового запису

Посилання: <https://github.com/NotWizaRD12/APZ>

## Практична робота №3

**Тема:** Написання тест-кейсів (Test Case).

**Мета:** Набування навичок у написанні тест-кейсів різних пристроїв.

Очікувані результати навчання: уміння підписувати особисту документацію з використанням єдиного цифрового підпису за допомогою різних сервісів і додатків. Короткі теоретичні відомості

**Тест-кейс** – це професійна документація тестувальника, це послідовність дій, спрямована на перевірку будь-якого функціоналу, що описує як прийти до очікуваного результату.

**Тема кейса** – описова назва тесту, яка спрощує його пошук та розуміння його змісту. У темі тест-кейсі не повинно бути залежностей від інших тест-кейсів і також нечітких формулювань та зайвої деталізації.

Навички, необхідні для створення тест-кейсів:

- **Збір і аналіз вимог до продукту.** Якщо формально описаних вимог (специфікацій) немає, тестувальник повинен уміти отримати необхідну інформацію від розробників, аналітиків чи користувачів.
- **Декомпозиція системи.** Необхідно не лише бачити систему загалом, а й уміти розділяти її на окремі складові. Ця навичка особливо корисна під час функціонального тестування, коли перевіряються всі частини продукту.
- **Уміння визначати пріоритети.** Тест-дизайнер має вміти відрізняти більш важливі аспекти від менш значущих і правильно розставляти пріоритети під час тестування.
- **Чітке формулювання думок (усно й письмово).** Це вміння є важливим для будь-якого тестувальника, а для тест-дизайнера — особливо під час опису тест-кейсів.

- **Знання технік тест-дизайну та їх практичне застосування.**

Володіння різними методами створення тестів і вміння ефективно використовувати їх у роботі.

Результатом виконання тест кейсу можуть бути:

- Позитивний результат (pass) – фактичний результат дорівнює очікуваному результату.
- Негативний результат (fail) – фактичний результат не дорівнює очікуваному результату. У цьому випадку знайдено помилку.
- Виконання тесту заблоковано (blocked) – після одного з кроків продовження тесту неможливе. У цьому випадку також знайдено помилку.

### **Важливі моменти при складанні тест-кейса:**

#### **Розуміння вимоги, за якою складається кейс**

Скласти й описати ефективний тест-кейс не вийде, якщо немає розуміння, яким саме чином повинен бути реалізований той чи інший функціонал програми. Ефективніше буде витратити сили та час на уточнення неточностей і в підсумку написати «робочий» тест-кейс, ніж спершу скласти абсолютно неправильний, зрозуміти в процесі тестування, що він не спрацював як треба, знову витратити час на уточнення, і тільки після цього повністю переробити та отримати робочий варіант.

#### **Простота та легкість для розуміння**

Важливо, щоб тест-кейс був описаний зрозумілими словами, без використання спеціальної термінології та складних мовних конструкцій, так як його можуть проходити фахівці різного рівня кваліфікації та спеціалізації (досвідчені тестувальники та початківці, керівники проекту,

розробники, аналітики та навіть співробітники замовника, не завжди володіють достатнім рівнем знань у галузі комп'ютерних технологій).

### **Окрема увага приділяється наборам тестових даних**

Якщо тест передбачає використання будь-яких даних, слід звернути увагу на правильність їх складання та подальші вказівки. В іншому випадку такий тест є марним.

### **Завдання.**

Придумайте об'єкт тестування. Це повинен буде будь-який об'єкт, який складається мінімум з 5 частин. (годинник, скейт, велосипед, монітор і т.п.) Напишіть не менше 20 тест кейсів до цього об'єкту, що зможуть оцінити його якість з точки зору продукту. Створіть \*.pdf документ, де кратко опишіть ваш об'єкт та потім опишіть ваші тест-кейси.

### **Об'єкт тестування: Велосипед**

#### **Опис об'єкта:**

Велосипед — це транспортний засіб, який складається з рами, коліс, керма, сидла, педалей, гальмівної системи та трансмісії. Його основна функція — забезпечення зручного та безпечного пересування користувача. Метою тестування є перевірка функціональності, надійності, зручності та безпеки експлуатації велосипеда.

#### **Тест-кейси**

№	Назва тест-кейсу	Опис перевірки	Очікуваний результат
1	Перевірка цілісності рами	Огляд на наявність тріщин чи пошкоджень	Рама без пошкоджень
2	Перевірка роботи переднього гальма	Натискання ручки гальма	Колесо повністю зупиняється
3	Перевірка роботи заднього гальма	Натискання заднього гальма	Колесо зупиняється без ривків
4	Перевірка обертання	Крутіння педалей вручну	Педали обертаються без

	педалей		заїдань
5	Перевірка стану ланцюга	Огляд на знос або іржу	Ланцюг цілий, без іржі
6	Перевірка перемикання передач	Перемикання всіх передач	Передачі перемикаються плавно
7	Перевірка висоти сидла	Регулювання сидла	Сідло легко регулюється і фіксується
8	Перевірка керма	Поворот уліво/управо	Кермо обертається вільно, без люфту
9	Перевірка дзвінка	Натискання дзвінка	Звук чіткий і гучний
10	Перевірка фар	Увімкнення фар	Фари працюють, світять рівномірно
11	Перевірка відбивачів	Огляд усіх відбивачів	Всі відбивачі цілі і чисті
12	Перевірка наявності підшипників	Огляд у втулках коліс	Підшипники присутні та працюють плавно
13	Перевірка тиску в шинах	Вимірювання манометром	Тиск відповідає нормі виробника
14	Перевірка стану шин	Огляд протектора	Шини без тріщин і проколів
15	Перевірка стійкості велосипеда	Поставити велосипед на рівну поверхню	Стоїть стійко, не хитається
16	Перевірка ланцюгового приводу	Прокрутка коліс і педалей	Привід працює без ривків
17	Перевірка підніжки	Виставити підніжку	Підніжка утримує велосипед
18	Перевірка гальмівних колодок	Огляд колодок	Колодки не зношені більше ніж на 50%
19	Перевірка звуку при русі	Коротка поїздка	Немає сторонніх шумів
20	Перевірка фарбового покриття	Візуальний огляд	Фарба без подряпин і відшарувань

## Практична робота №4

**Тема:** AWS S3.

**Мета:** Набування навичок у створення і розміщенні статичної веб-сторінки на AWS S3.

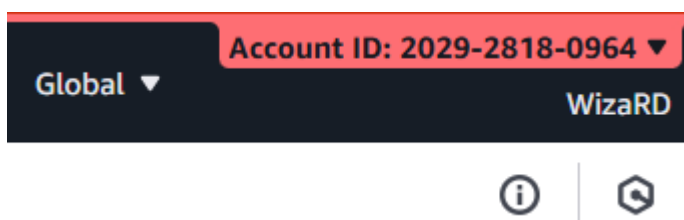
**Очікувані результати навчання:** уміння створити і розмістити сторінку з власними даними на ресурсі AWS S3.

### Короткі теоретичні відомості

**Amazon S3 (Simple Storage Service)** — це сервіс від компанії Amazon Web Services для зберігання об'єктів (файлів) у хмарі. Він дозволяє зберігати будь-який тип даних: документи, зображення, відео, резервні копії, веб-сторінки тощо.

AWS S3 — це гнучкий, потужний та простий сервіс для зберігання даних, який також можна використовувати для хостингу простих статичних сайтів. Він дозволяє завантажити HTML-файл і отримати публічну URL-адресу для доступу через інтернет.

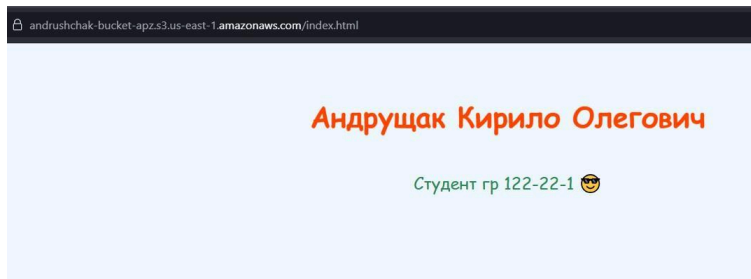
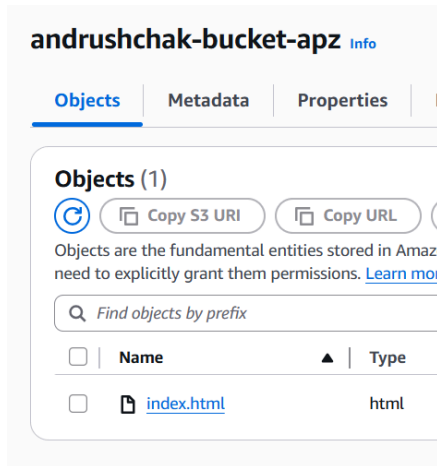
Крок 1. Зареєструватися в системі AWS.



Крок 2. Створити бакет у S3 з вашим прізвищем та іменем.

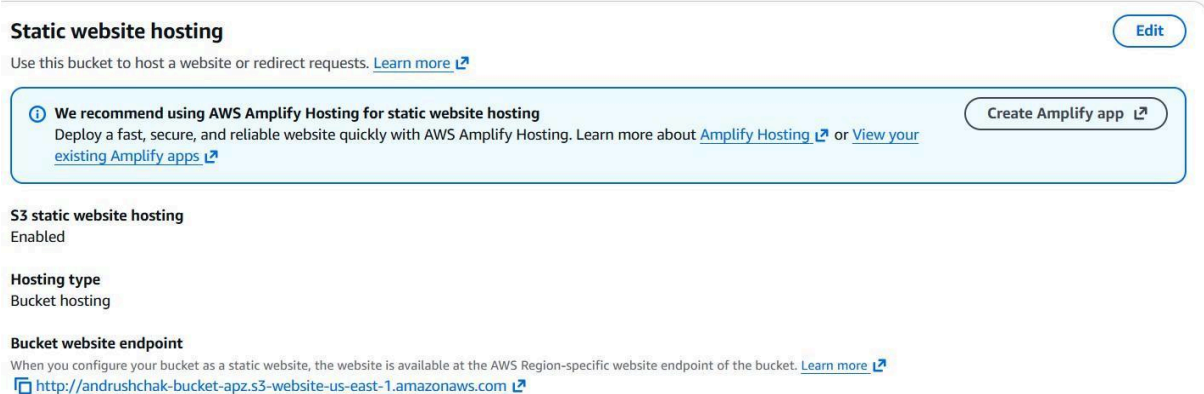
✔ Successfully created bucket "andrushchak-bucket-apz"  
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Крок 3. Розмістити на S3 статичну веб-сторінку, яка містить ваше ПБ та Вашу академічну групу.



Крок 4. Налаштування хостингу і отримання публічної адреси сторінки, наприклад:

<https://khaleiko-bucket-anz.s3-website-eu-north-1.amazonaws.com/>



<http://andrushchak-bucket-apz.s3-website-us-east-1.amazonaws.com>



## Практична робота №5

**Тема:** AWS S3.

**Мета:** набування навичок створення та розміщення віртуального сервера за допомогою AWS EC2.

**Очікувані результати навчання:** уміння створити і розмістити сторінку з власними даними на ресурсі AWS EC2.

### Короткі теоретичні відомості

**Amazon EC2 (Elastic Compute Cloud)** — це хмарний сервіс компанії Amazon Web Services, який дозволяє створювати та запускати віртуальні сервери (EC2-інстанси) у хмарі. Він забезпечує гнучкі обчислювальні ресурси, які можуть масштабуватися відповідно до потреб користувача.

EC2 дає можливість:

- створювати віртуальні машини з різними характеристиками (процесор, оперативна пам'ять, диск);
- обирати операційну систему (Linux, Windows тощо);
- налаштовувати мережу, безпеку та доступ до інстанса; розгорнути веб-сайти, серверні додатки, бази даних та інші сервіси;
- оплачувати лише використані ресурси (погодинна або секунда за секундами тарифікація).

EC2 використовується для хостингу сайтів, бекенд-додатків, тестових середовищ, ігрових серверів та будь-яких інших задач, що потребують доступного та масштабованого обчислювального ресурсу.

## Крок 1. Створюємо та запускаємо Instance

☰ [EC2](#) > [Instances](#) > [Launch an instance](#)

📘 It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances, and how to manage them.

[Take a walkthrough](#) [Do not show me this message again.](#)

### Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the steps below.


#### Name and tags [Info](#)


**Name**


[Add additional tags](#)


Рисунок 1.1. Name and tags


#### Quick Start


Amazon Linux  



macOS  


Ubuntu  


Windows  


Red Hat  


SUSE Linux  


Debian  


>

Рисунок 1.2. Amazon machine Image

**Key pair name**  
Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It must start with a letter and end with a letter or digit.

**Key pair type**

☒ RSA  
RSA encrypted private and public key pair

**Private key file format**

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

Рисунок 1.3. Створення key pair

▼ **Configure storage** [Info](#) Advanced

1x  GiB  Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

🔄 Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

Рисунок 1.4. Налаштування Configure Storage

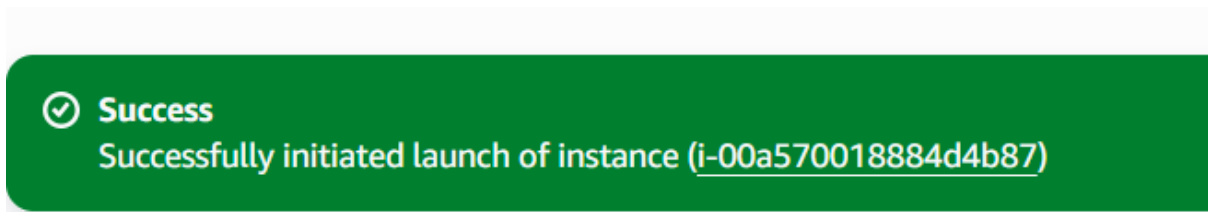


Рисунок 1.5. Успішне створення Instance

## Крок 2. Отримання зашифрованого паролю

**Get Windows password** [Info](#)

Use your private key to retrieve and decrypt the initial Windows administrator password for this instance.

**Instance ID**  
 (Andrushchak-apz-122-22-1)

**Key pair associated with this instance**

**Private key**  
Either upload your private key file or copy and paste its contents into the field below.

[Upload private key file](#)

Private key contents

**Public DNS**

**Password**

Рисунок 2.1-2.2. Windows password

### Крок 3. Підключаємося до створеного ПК

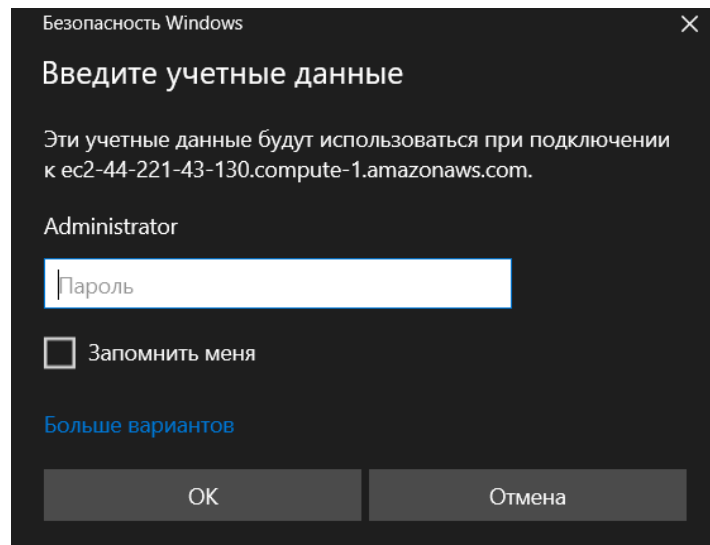


Рисунок 3.1 Windows password

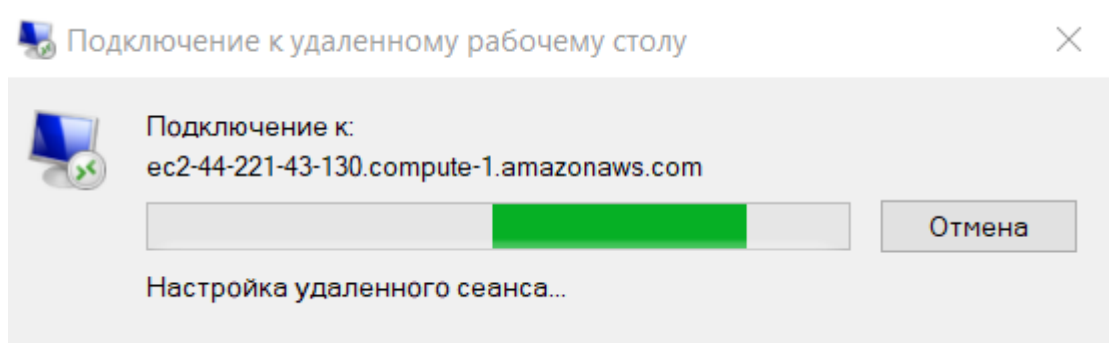


Рисунок 3.2 Завантаження EC2 Instances

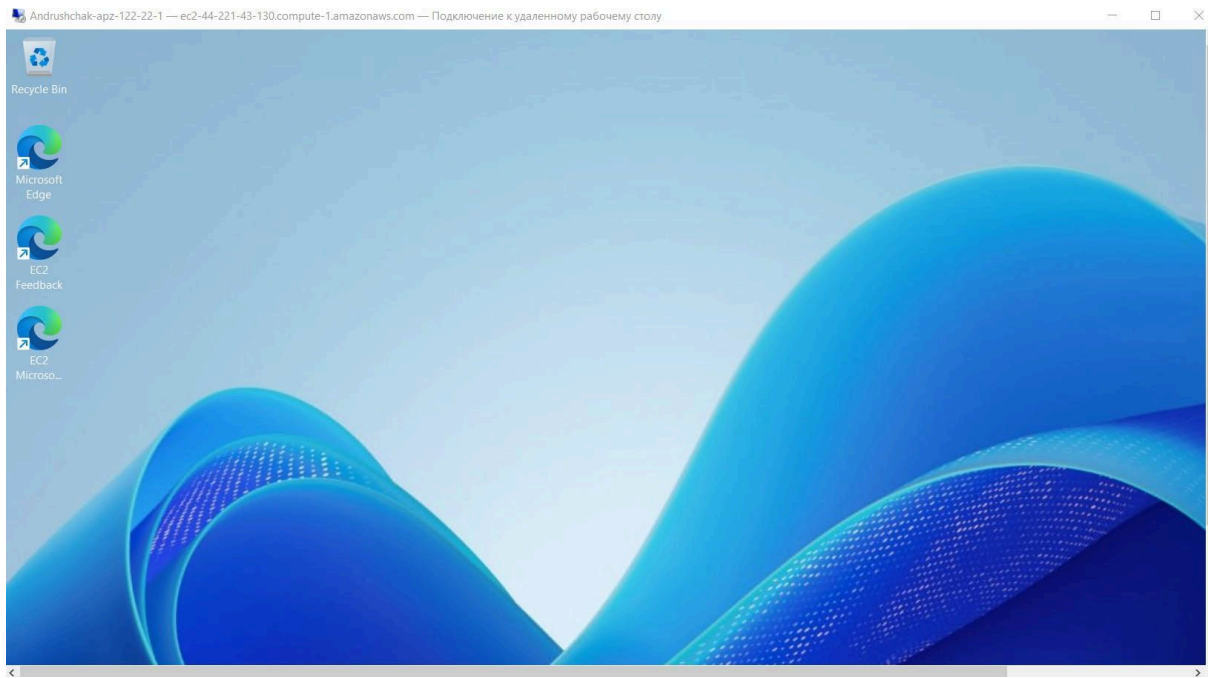


Рисунок 3.3 Відображення робочого столу