WILLIAM OLOO OKANGA

2100719370

21/U/19370/PS

a)In the first class , i.e ‘Calculation 1’, the constructor method initiates two instance variables ‘x1’ and x2 and assigns them the values x and y respectively. Within the same Calculation 1 class , a method Sum is defined that prints the values of x1 and x2 and returns their sum.

In the second class , ‘Calculation 2’ , a method named Subtraction is defined that returns the value of subtracting the value of x2 from the value of x1. Within the same class 2 other methods Multiplication and Division are defined that return the result of multiplying x1 with x2 and the result of dividing x1 by x2 respectively.

The final class in the code snippet is the Derived class that inherits from the previous 2 classes. The class definition is empty hence the pass statement.

The remaining part of the code takes user input for the values of x and y and creates an instance of the derived class using these 2 values. We then call the sum method on this instance and print the result .

The last two numbers of my student number are 7 and 0 , hence the output is 7.

b)

Maintaining Existing Functionality:

Regression testing's primary purpose is to guarantee that modifications to the codebase do not negatively effect current functionality. This is critical in a data processing module when processing data correctness and dependability are critical. Regression testing aids in the preservation of existing functionality.

Adapting to Changes:

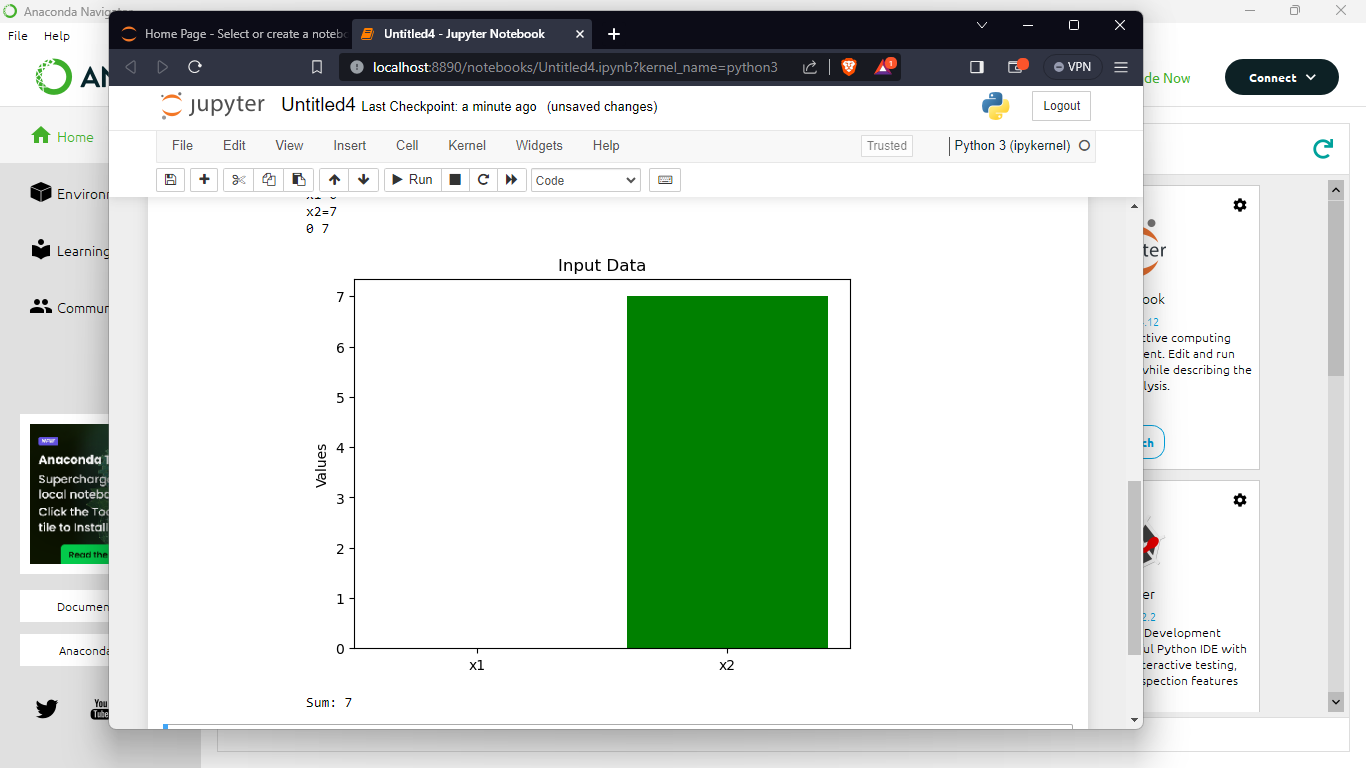
Changes to the codebase are frequently involved in updates and enhancements. The development team can use regression testing to ensure that the changes do not break existing code. It aids in modifying the code to new requirements without jeopardizing the module's stability.

Maintaining Cross-Functionality:

Different components in a data processing module may be interconnected. Regression testing ensures that changes to one portion of the module do not have an unfavorable effect on other components or functionalities. It contributes to the system's overall cross-functionality.

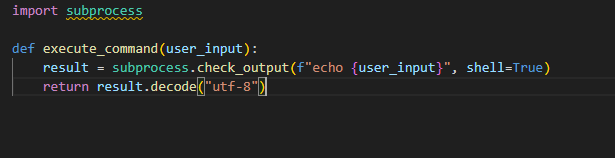
CI/CD (Continuous Integration and Deployment):

Regression testing is an essential component of the pipeline in an environment where continuous integration and deployment procedures are used. Automated regression tests can be integrated into the CI/CD process to ensure that every code change is thoroughly verified before deployment, reducing the likelihood of bugs entering production.

c)

2.

**OS Command Injection**



This code accepts user input (user\_input) and utilizes it directly in a shell command without any validation or sanitization. An attacker might inject malicious commands, allowing the underlying operating system to execute arbitrary commands.

**Cross-site Scripting (XSS):**

A screen shot of a computer

Description automatically generated

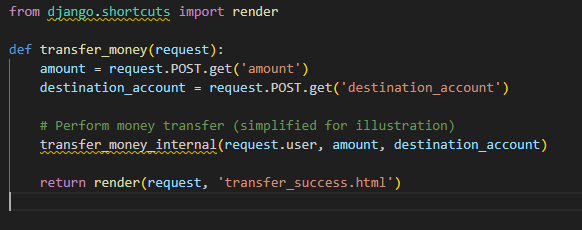
In this example, the user input (message) is directly rendered in the HTML template without proper escaping. An attacker could inject malicious scripts into the message parameter, leading to the execution of unauthorized scripts in the user's browser.

**Session Hijacking:**



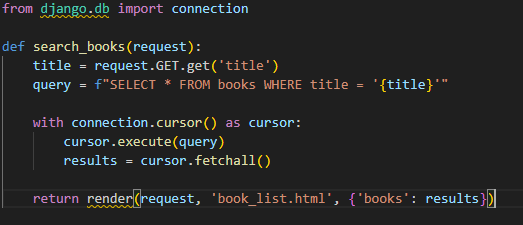
The user's session ID is stored in the session without sufficient protection in this case. An attacker can hijack a session and impersonate the user if they obtain or guess a legitimate session ID.

**Cross-Site Request Forgery (CSRF):**



The CSRF token is not correctly validated in this scenario. By creating a malicious website that submits queries to the target site on behalf of the user, an attacker could fool a user into doing an unpleasant activity, such as transferring money.

**SQL Injection:**



The title parameter is directly put into the SQL query in this example, without any sanitization or parameterization. An attacker might use the title argument to inject malicious SQL code, potentially resulting in unauthorized database access or manipulation.

Use Django's built-in security mechanisms, such as: to safeguard a Django project against these vulnerabilities.

For database interactions, parameterized queries are used.

Django forms are used to handle user input and validation.

Django's template framework for HTML escaping automatically.

Django's authentication system for securely handling user sessions.

Django's CSRF protection is used to prevent CSRF attacks.

Always adhere to web application security best practices, keep software libraries up to date, and conduct frequent security audits to discover and remediate potential vulnerabilities.