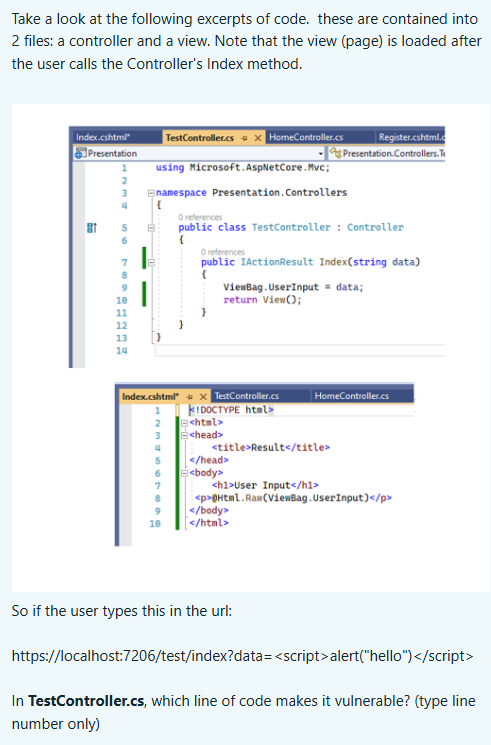
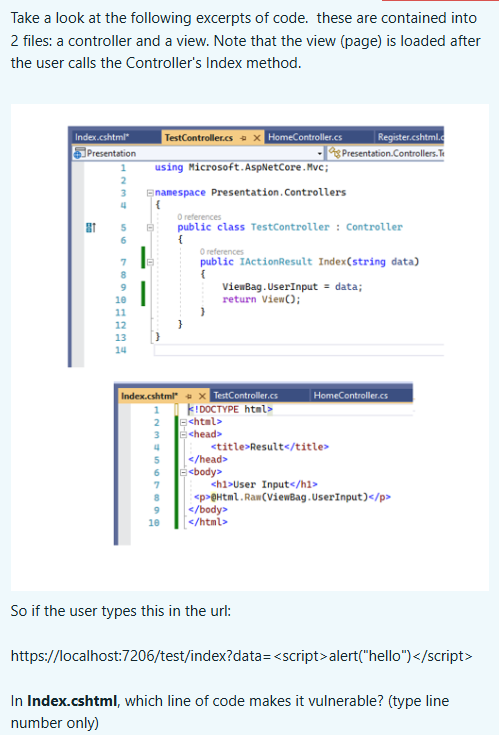
Securing Applications time constrained assignment

**Question 1:**



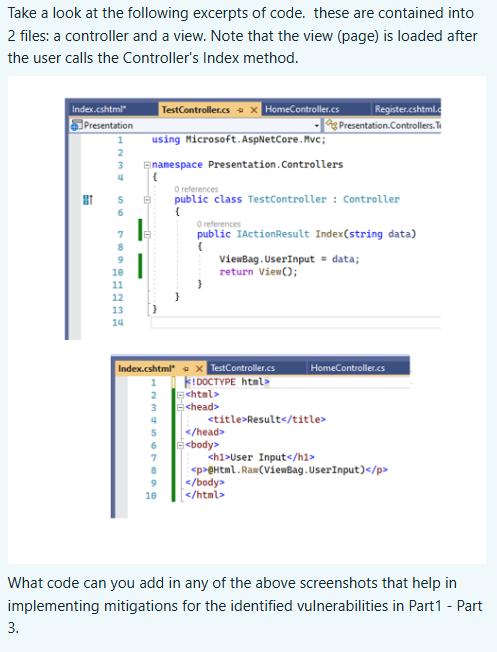
**Question 2:**



**Question 3:**



**Question 4:**



**Question 5:**

The question provides the following information: a cipher; a key; and an IV.

Key: 104 193 16 25 27 37 83 229 33 37 178 143 128 51 217 15 221 138 82 40 150 114 167 212 203 85 235 247 165 81 22 204

IV: 235 240 99 118 142 204 17 97 98 153 22 91 77 66 103 242

Cipher: v4WAkA6Prb1GJ8/OEXJLTg==

The cipher is to be decrypted using an AES symmetric key encryption from the System.Security.Cryptography namespace. These are things to take note prior to start developing the symmetric decryption method:

Algorithm:AES

Cipher Mode: ECB

Padding Mode: ANSIX923

The answer which will be a string is a 2 word human readable phrase with no symbols or nonsense characters [2]

**Question 6:**

Paste your decryption code here.

Here is how marks will be split:

Create a Decrypt method with the following signature: string Decrypt(string plaintext)

Create an instance of the AES algorithm with the appropriate Key, IV, CipherMode and PaddingMode. The Key and IV are bytes separated by a space to be converted to byte[] (1 mark);

Decode the cipher from base64 string to obtain an array of bytes (1 mark);

Create a decryption cryptographic stream that can consume the byte array and provide the decrypted bytes. (3 marks)

**Question 7:**

A misconfigured server allows unauthorized users to view internal logs, which include administrator passwords in plaintext.

Select one:

a. Information Disclosure

b. Spoofing

c. Elevation of Privilige

d. Tampering

e. Denial of Service

f. Repudiation

**Question 8:**

A user's session token is intercepted and used to impersonate them on a web application.

Select one:

a.Elevation of Privilige

b.Repudiation

c.Spoofing

d.Information Disclosure

e.Denial of Service

f.Tampering

**Question 9:**

An attacker overwhelms a company’s customer service chatbot using automated requests, disrupting access for legitimate users.

Select the option that fits the scenario from the list below

Select one:

a. Tampering

b. Spoofing

c. Repudiation

d. Denial of Service

e. Elevation of Privilige

f. Information Disclosure

**Question 10:**A script injection vulnerability is exploited to alter a user's financial transactions on a banking website

Select the option that fits the scenario from the list below

Select one:

a. Spoofing

b. Denial of Service

c. Tampering

d. Elevation of Privilige

e. Information Disclosure

f. Repudiation

**Question 11:**

A vulnerability in the authentication mechanism allows users to execute privileged commands without proper authorization.

Select the option that fits the scenario from the list below

Select one:

a. Information Disclosure

b. Denial of Service

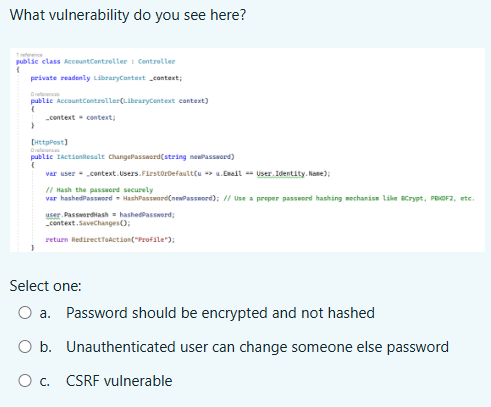
c. Tampering

d. Spoofing

e. Elevation of Privilige

f. Repudiation

**Question 12:**



**Question 13:**

So after having identified the vulnerability here:

What do you propose applying as a mitigation:



Select one:

a. Checking that the user knows the old password

b. Anti CSRF attribute on top of the method

c. Implement an ActionFilter validating the user authenticity

**Question 14:**



**Question 15:**



**Question 16:**



**Question 17:**

Which OWASP category does this vulnerability fall under?

|  |  |
| --- | --- |
| A user is able to access another user’s profile page by simply changing the user ID in the URL from their own to someone else's. For example, changing /profile/12345 to /profile/67890 allows them to view another user’s sensitive data. | Answer 1 |
| A server-side application parses XML files uploaded by users. However, the XML parser is not properly configured to disable external entity processing. An attacker uploads a malicious XML file that references an external entity, allowing the attacker to read sensitive files from the server. | Answer 2 |
| An e-commerce website stores credit card numbers in plain text in its database. The application sends these card numbers over an insecure HTTP connection without encryption. A malicious attacker can intercept the data, gaining access to customers' sensitive payment details. | Answer 3 |
| A web application allows users to authenticate using only their username and password. After login, the session is maintained using a weak, predictable session token that is stored in a cookie. An attacker can guess the session token value and gain unauthorized access to another user's account. | Answer 4 |
| A website allows users to input a search term into a text box.  The application directly inserts this input into a SQL query to fetch records without proper sanitization. A user inputs the following: ' OR 1=1 -- This causes the query to return all records from the database, bypassing authentication. | Answer 5 |

**Question 18:**

You are provided with four files:  [click here to download](https://vle.mcast.edu.mt/pluginfile.php/148166/question/questiontext/43259/18/55659/keys2.zip)

1.      **privatekey.txt** - This file contains a private RSA key in Base64 format generated by who?.

2.      **publickey1.txt** - A file containing an RSA public key in Base64 format generated by Alice.

3.      **publickey2.txt** - A file containing another RSA public key in Base64 format generated by Tom.

4.      **publickey3.txt** - A file containing yet another RSA public key in Base64 format generated by Bob.

**Task:**  
Determine which one of the three public keys (publickey1.txt, publickey2.txt, publickey3.txt) corresponds to the private key in privatekey.txt.

You must write which public key (**by writing down in the textbox provided the name of the person attached**) is the correct one to verify the authenticity of the data signed by the private key.  Save the code because it will have to be uploaded in a separate question.  Part of the code is provided to you to assist you in building the right method that can determine the answer to the question – you need only code lines with comments //Answer1 & //Answer2  Note: you might need to use SHA256 (for hashing) AND pkcs1 (as a signature padding)

private bool VerifyKeyPair(byte[] privateKeyBytes, byte[] publicKeyBytes)  
 {  
 bool foundMatchingPublicKey = false;  
 try  
 {  
 using (RSA privateKeyRsa = RSA.Create())  
 using (RSA publicKeyRsa = RSA.Create())  
 {  
 privateKeyRsa.ImportRSAPrivateKey(privateKeyBytes, out \_);  
 publicKeyRsa.ImportRSAPublicKey(publicKeyBytes, out \_);  
 // Test the keys by signing and verifying a message  
 byte[] testMessage = Encoding.UTF8.GetBytes("TestMessage");  
 //Answer1: sign testMessage here  
 //Answer2: verify signature produced here  
 }  
 }  
 catch  
 {  
 return false; // If any error occurs, the keys do not match  
 }  
 }

**Question 19:**

This is the part where you need to submit your code for question 18 as presented to you earlier in this TCA.

Note: You had to replace comments //Answer 1 & //Answer 2.

each correct line carries 2 marks

**Question 20:**

Implement a method to verify whether the contents of file (download here) were actually tampered since this assignment was written. Digest produced using SHA256 at the time of writing the assignment was this:

vniFoU8cYTPLrO7eqi4IHOBPSCQSdPVe4tn5+SXJOJc=

What’s your conclusion, was it tampered? [2]

Select one:

True

False

**Question 21:**

Submit the method or lines of code implemented to answer Part 1 (of the previous question)

**Question 22:**

