**ASSIGNMENT – 4**

**STEP-1:**

**OWASP TOP 10 VULNERABILITIES OVERVIEW :**

1. **Broken Access Control (BAC):** BAC vulnerabilities occur when an application grants access to unauthorized users or resources. This can be caused by a variety of factors, such as misconfigured access control lists (ACLs), insecure coding practices, and human error.
2. **Cryptographic Failures:** Cryptographic failures occur when an application uses weak or insecure cryptographic algorithms, keys, or protocols. This can allow attackers to eavesdrop on sensitive communications, tamper with data, or even impersonate legitimate users.
3. **Injection:** Injection vulnerabilities occur when an application allows attackers to inject malicious code into data that is processed by the application. This can be used to steal data, hijack user sessions, or even take control of the application server.
4. **Insecure Design:** Insecure design flaws are weaknesses that are inherent in the design of an application. These flaws can be very difficult to fix, and they can make the application vulnerable to a wide range of attacks.
5. **Security Misconfiguration:** Security misconfiguration vulnerabilities occur when an application is not configured securely. This can be caused by a variety of factors, such as using default settings, failing to apply security patches, and misconfiguring security features.
6. **Vulnerable and Outdated Components:** Vulnerable and outdated components are third-party libraries or frameworks that contain known security vulnerabilities. These vulnerabilities can be exploited by attackers to compromise the application.
7. **Identification and Authentication Failures:** Identification and authentication failures occur when an application does not properly identify or authenticate users. This can allow attackers to impersonate legitimate users or gain access to unauthorized accounts.
8. **Software and Data Integrity Failures:** Software and data integrity failures occur when an application does not ensure the integrity of its software and data. This can allow attackers to tamper with data or code, which can have a variety of negative consequences.
9. **Security Logging and Monitoring Failures:** Security logging and monitoring failures occur when an application does not properly log security events or monitor for suspicious activity. This can make it difficult to detect and respond to security incidents.
10. **Server-Side Request Forgery (SSRF):** SSRF vulnerabilities occur when an application can be tricked into making unauthorized HTTP requests to a different server. This can be used by attackers to attack internal systems, steal data, or carry out other malicious activities.



* **OWASP top 10 vulnerabilitIes represent the most critical security risks to web applications, including injection attacks, broken authentication, sensitive data exposure, XML external entities (XXE), etc:**

1. **Injection Attacks (A03:2021):** Imagine a malicious actor sneaking hidden commands into user inputs like login forms or search bars. That's exactly what injection attacks are about. Attackers inject malicious code (like SQL code in SQL injection) that the application treats as legitimate data. This can wreak havoc, allowing attackers to steal data, manipulate databases, or even take control of the server.
2. **Broken Authentication (A07:2021):** This vulnerability exposes weaknesses in how applications identify and verify users. It could be weak password hashing, insecure session management, or flawed authorization controls. Attackers can exploit these weaknesses to gain unauthorized access to accounts, impersonate legitimate users, or bypass security measures altogether.
3. **Sensitive Data Exposure (Addressed through other categories):** While not a specific category anymore, the OWASP Top 10 recognizes the importance of protecting sensitive data. It tackles this through categories like "Cryptographic Failures" (A02:2021). This ensures applications use strong encryption algorithms and key management practices to safeguard sensitive information like credit card details or personal data.
4. **XML External Entities (XXE) (Not on the current OWASP Top 10 list):** XXE vulnerabilities were previously a concern, but they're not on the current OWASP Top 10 (2021) list. The focus has shifted towards broader categories like "Injection" which might encompass XXE in some scenarios. XXE vulnerabilities existed in XML processing and could allow attackers to exploit external entities to steal data or execute arbitrary code on the server.

## discuss the potential impact of these vulnerability on web application security and the importance of addressing them to prevent exploitation by attackers :

1. **Injection Attacks (A03:2021):** Imagine a malicious actor sneaking hidden commands into user inputs like login forms or search bars. That's exactly what injection attacks are about. Attackers inject malicious code (like SQL code in SQL injection) that the application treats as legitimate data. This can wreak havoc, allowing attackers to steal data, manipulate databases, or even take control of the server.
2. **Broken Authentication (A07:2021):** This vulnerability exposes weaknesses in how applications identify and verify users. It could be weak password hashing, insecure session management, or flawed authorization controls. Attackers can exploit these weaknesses to gain unauthorized access to accounts, impersonate legitimate users, or bypass security measures altogether.
3. **Sensitive Data Exposure (Addressed through other categories):** While not a specific category anymore, the OWASP Top 10 recognizes the importance of protecting sensitive data. It tackles this through categories like "Cryptographic Failures" (A02:2021). This ensures applications use strong encryption algorithms and key management practices to safeguard sensitive information like credit card details or personal data.
4. **XML External Entities (XXE) (Not on the current OWASP Top 10 list):** XXE vulnerabilities were previously a concern, but they're not on the current OWASP Top 10 (2021) list. The focus has shifted towards broader categories like "Injection" which might encompass XXE in some scenarios. XXE vulnerabilities existed in XML processing and could allow attackers to exploit external entities to steal data or execute arbitrary code on the server.

**STEP-2:**

**ALTORO MUTUAL WEBSITE ANALYSIS:**

**VULNERABILITY IDENTIFICATION REPORT:**

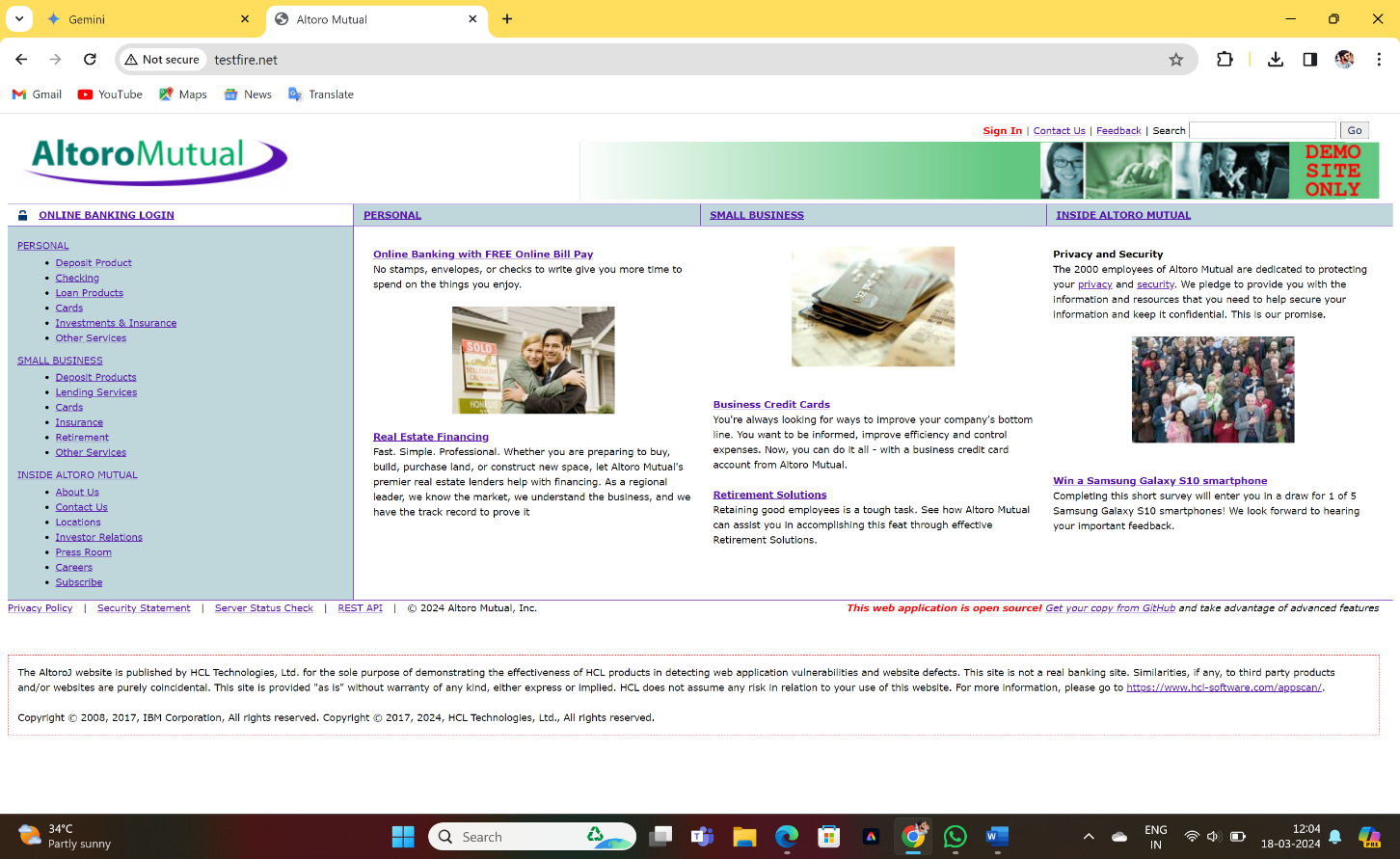
There are reports of an authorization issue (also known as IDOR - Insecure Direct Object Reference) vulnerability in the past regarding the Altoro Mutual website. This vulnerability allowed someone to potentially access information belonging to other users.

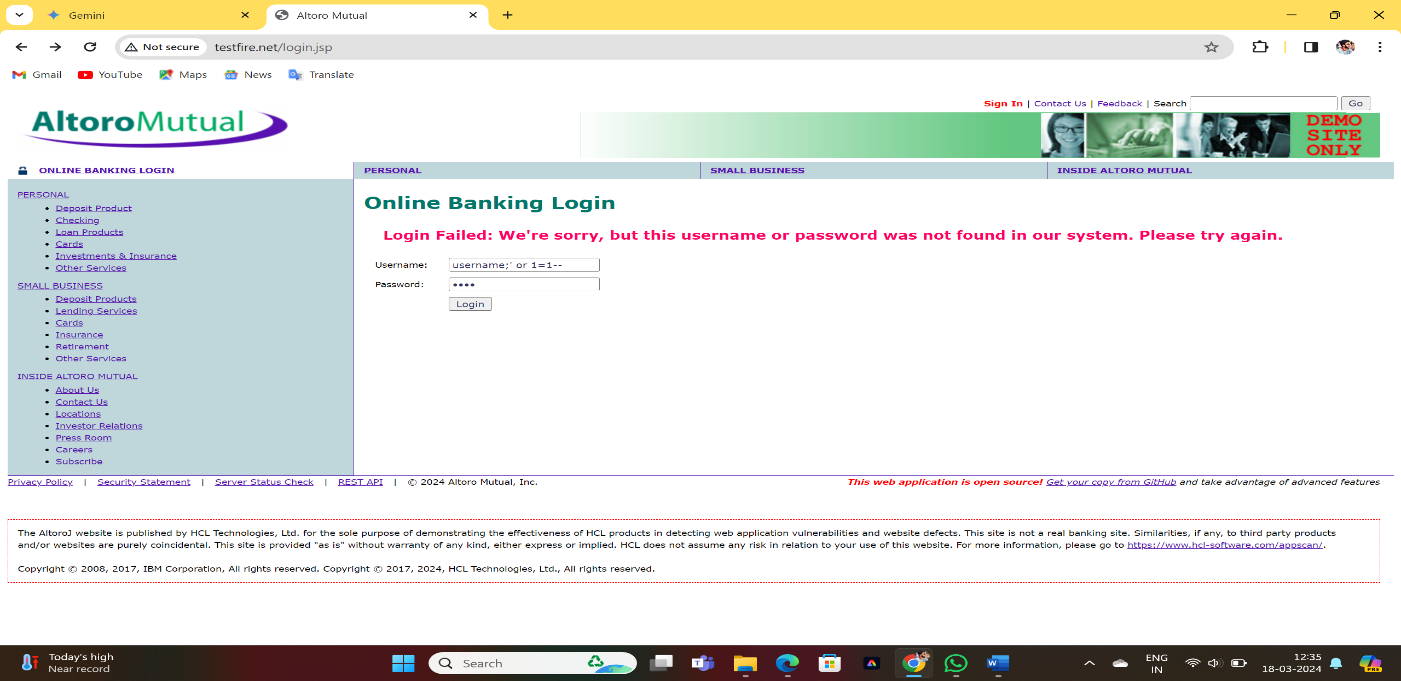
Here's what you should know:

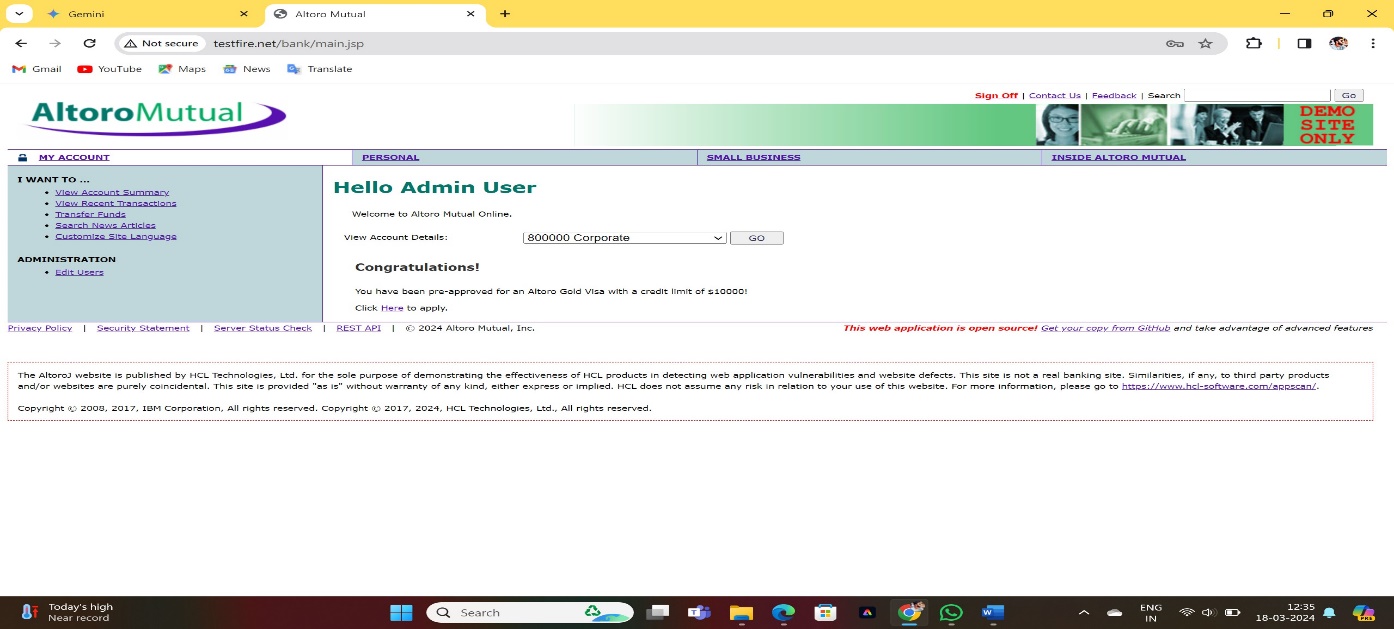
* This vulnerability seems to be from 2013 based on available information. Security practices and software development techniques evolve, so it's possible (and hopefully likely) that Altoro Mutual has addressed this issue.
* It's important to be aware of potential security risks, but also to rely on the company to address them.

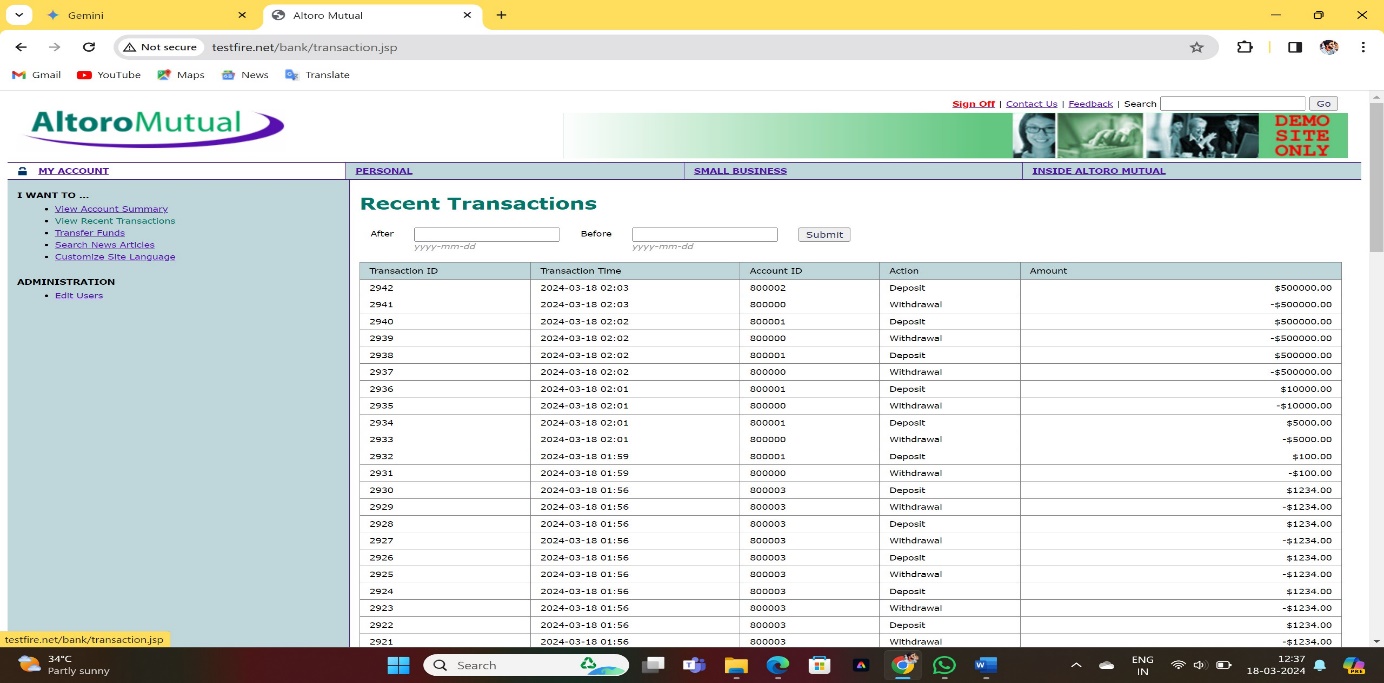
Here are some things you can do:

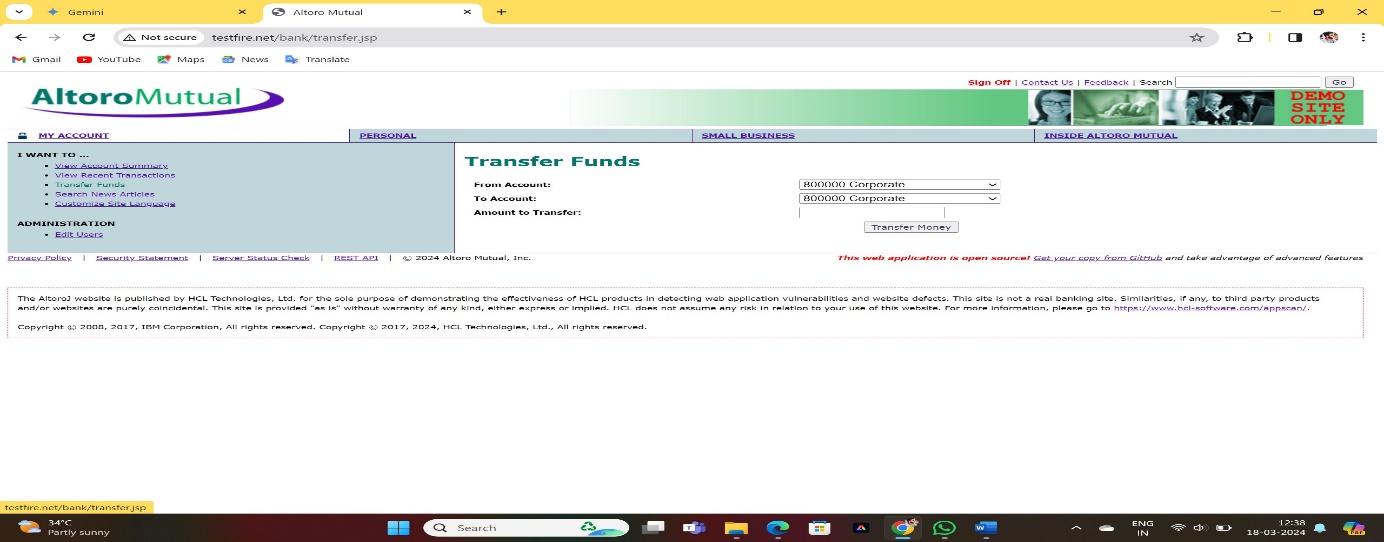
* **Check for Updates:** Financial institutions often notify users about security updates. See if Altoro Mutual has any recent security notices on their website.
* **Strong Passwords:** Use strong, unique passwords for your Altoro Mutual account and other financial accounts. Don't reuse passwords across different websites.
* **Be Phishing Aware:** Phishing emails try to trick you into revealing your login credentials. Be cautious of any emails claiming to be from Altoro Mutual and never click on links or attachments in such emails.

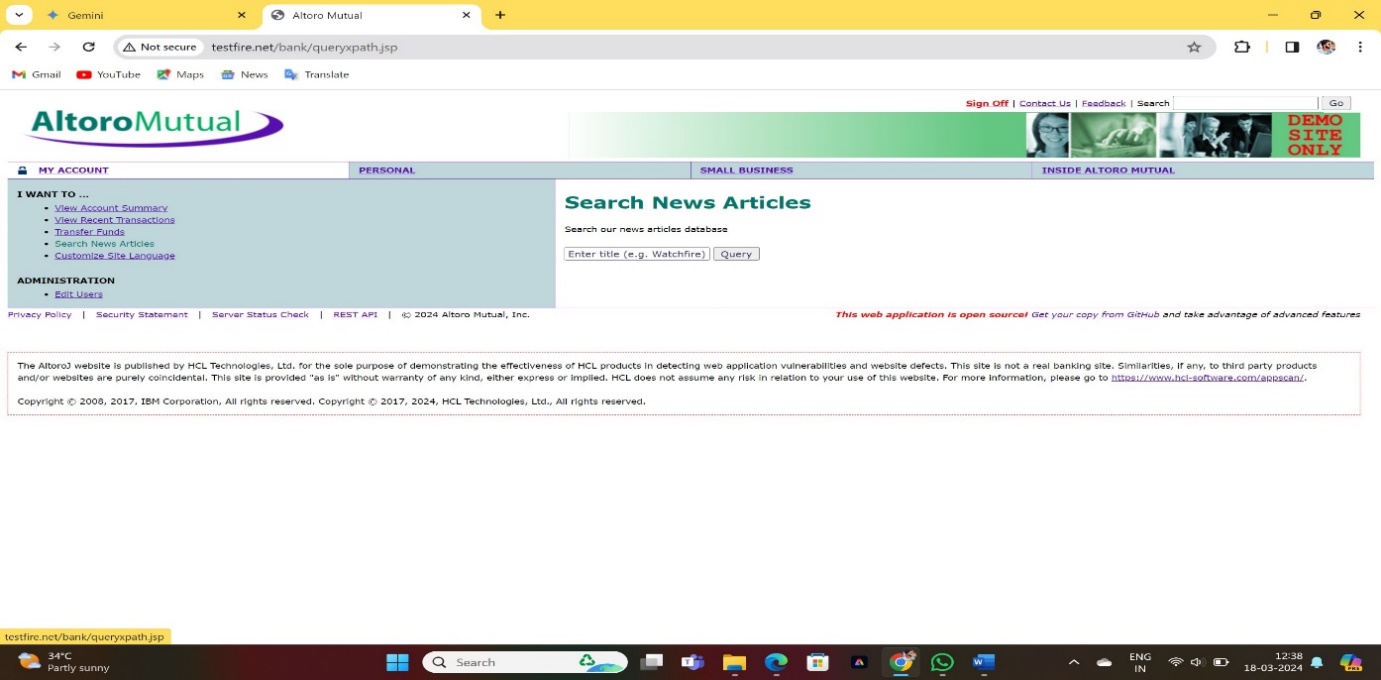
****

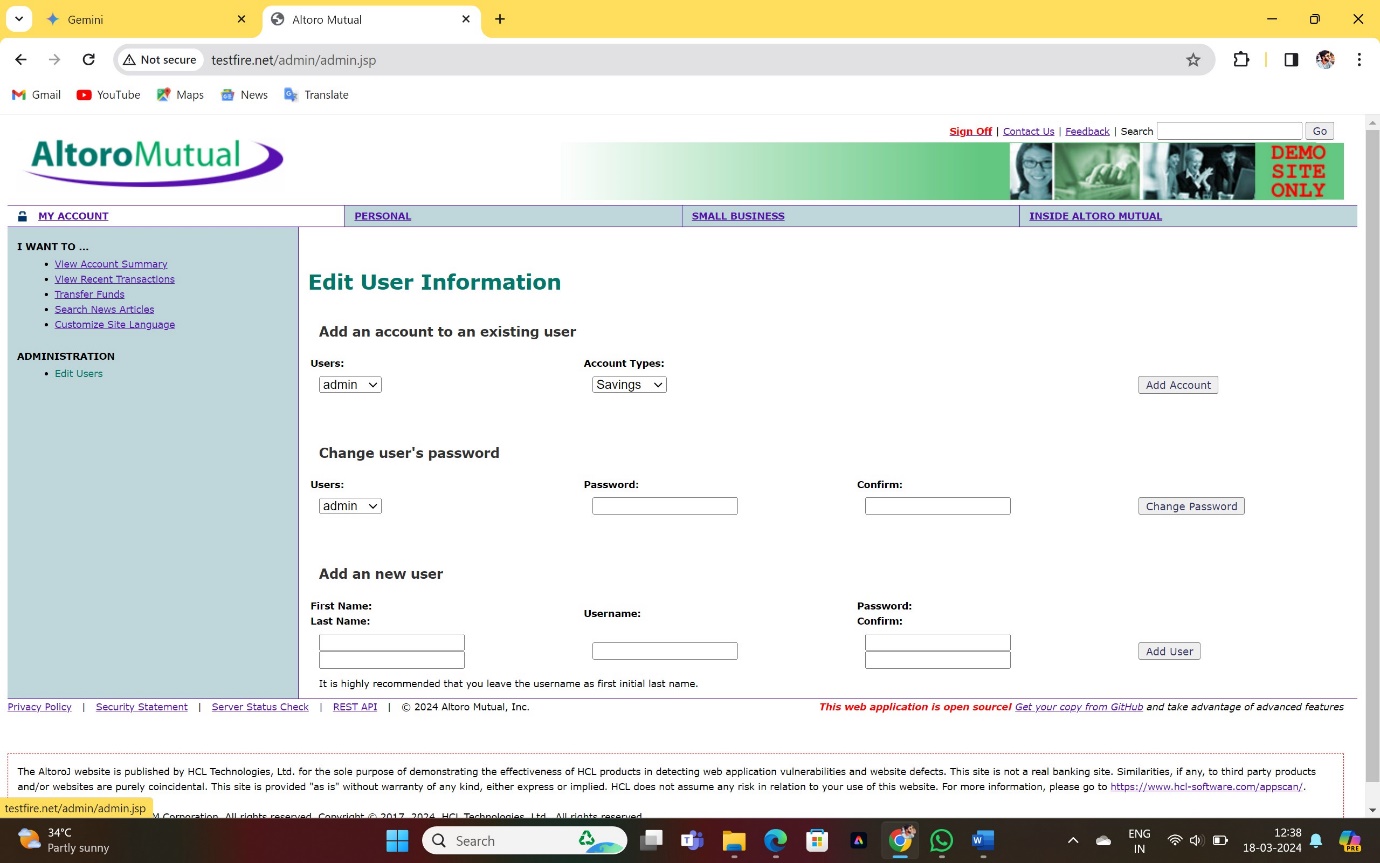
****

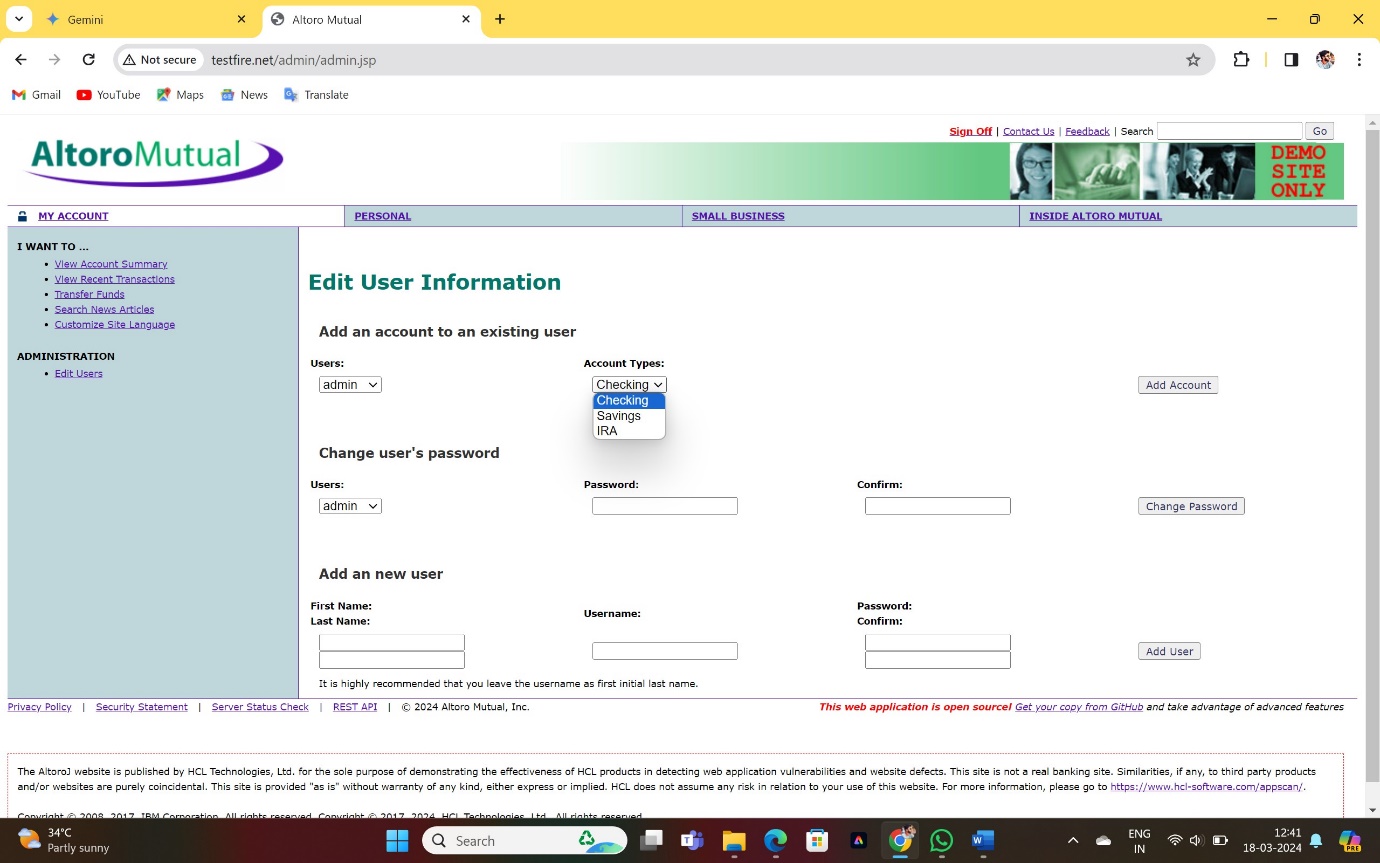










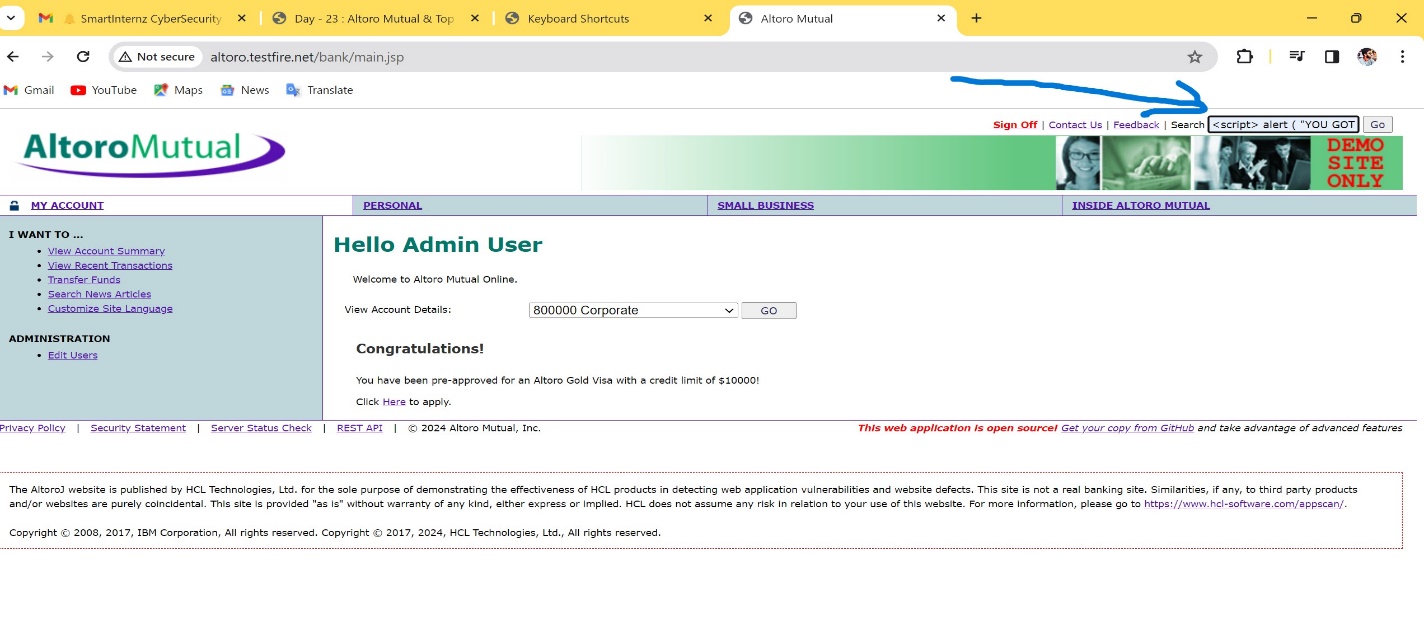


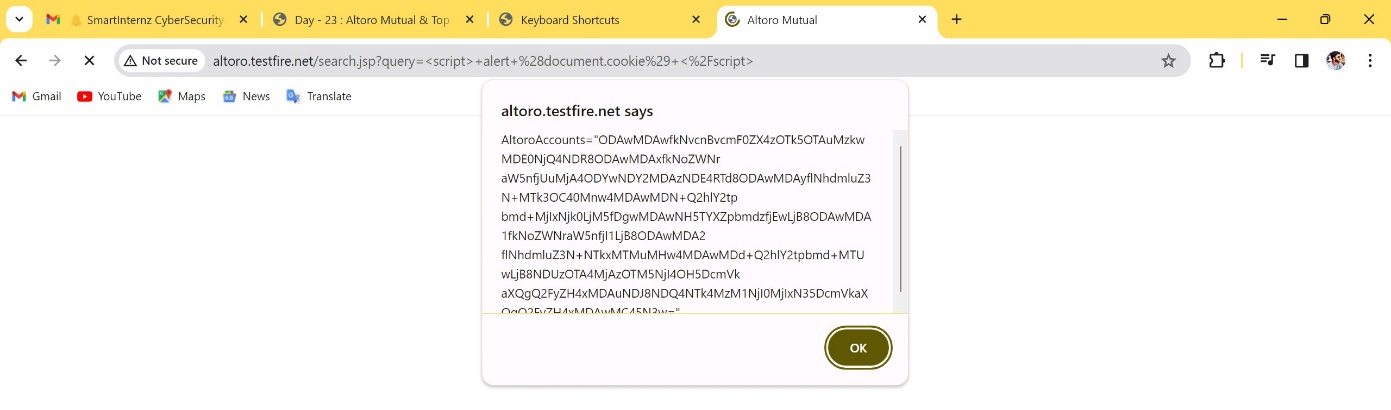
**CROSS - SITE - SCRIPTING :**

what you can do to stay secure:

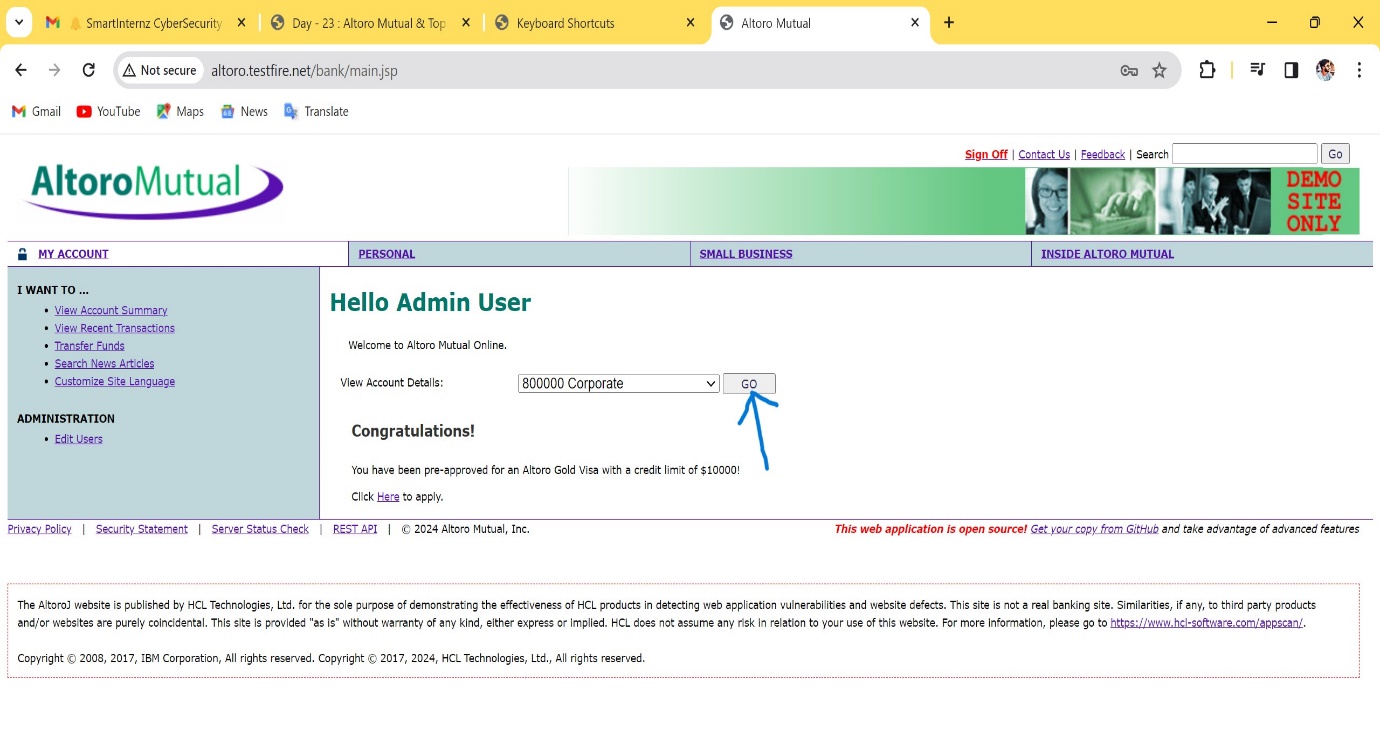
* **Altoro Mutual Updates:** Check the Altoro Mutual website for any security notices. They might announce updates that address these vulnerabilities.
* **Stay Skeptical:** Don't click on suspicious links or enter sensitive information on untrusted websites that look like Altoro Mutual. Phishing attacks can exploit XSS vulnerabilities.
* **Secure Browsing:** Keep your web browser and plugins updated to benefit from the latest security patches.

While the reported XSS vulnerabilities might be fixed, it's always good practice to be cautious. If you're very concerned, consider contacting Altoro Mutual's customer support to inquire about their security practices.

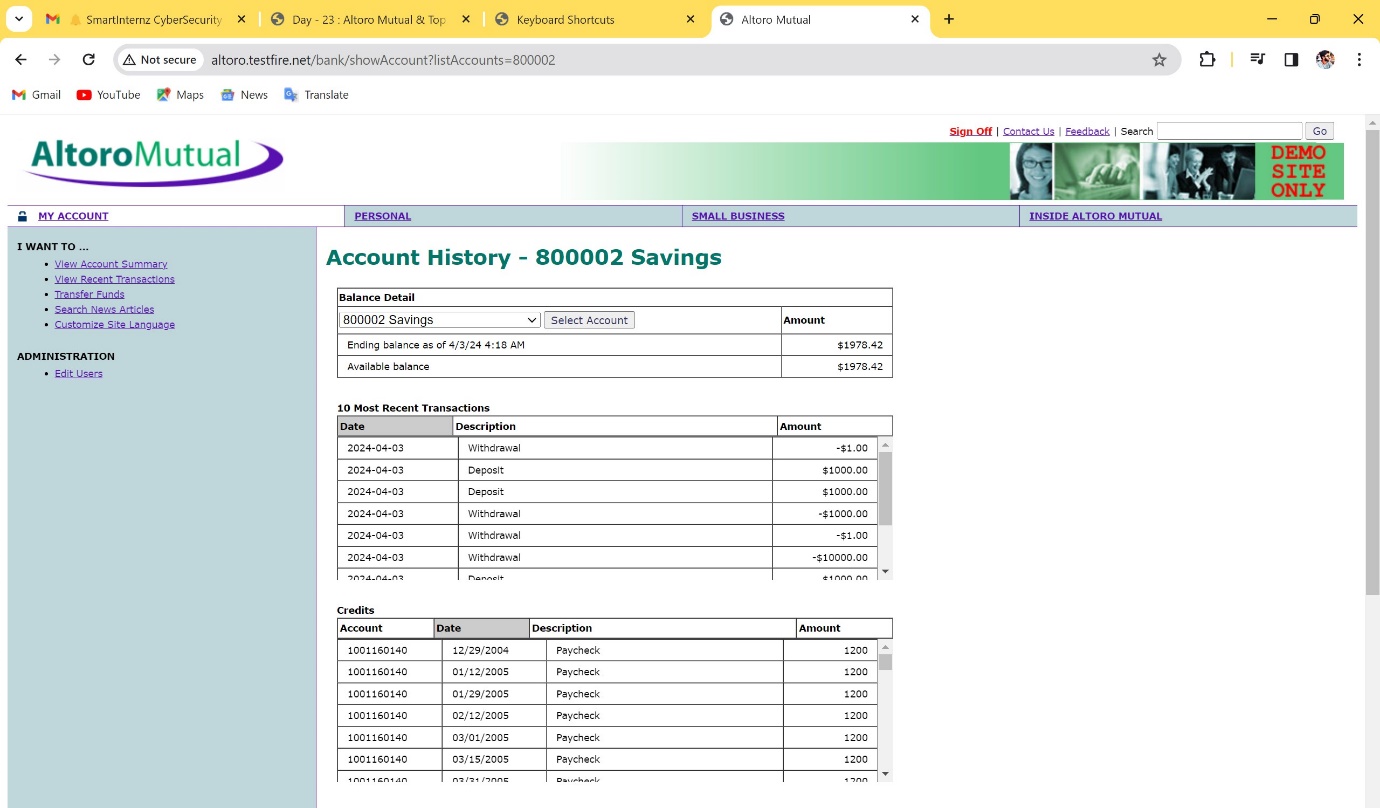




INSECURE DIRECT OBJECT REFERENCE :



OUTPUT :



**STEP-3:**

**MITIGATION STRATEGY PROPOSAL:**

**INPUT VALIDATION AND SANITIZATION:**

Validate and sanitize all user input on the server-side to ensure that it meets expected formats and does not contain malicious code

**CONTENT SECURITY POLICY (CSP):**

Implement a Content Security Policy (CSP) to specify which resources the browser should execute or load, thereby mitigating the execution of injected scripts.

**OUTPUT ENCODING:**

Encode user-generated content before rendering it in HTML context to prevent browsers from interpreting it as executable code.

**HTTPONLY AND SECURE COOKIES:**

Set the HttpOnly flag on cookies to prevent client-side scripts from accessing them, thereby mitigating the risk of stealing session tokens via XSS attacks.

**USE FRAMEWORKS WITH BUILT-IN XSS PROTECTION:**

Keep framework dependencies up-to-date to ensure that you benefit from the latest security enhancements and patches.

**CONTEXTUAL OUTPUT ESCAPING:**

Apply output encoding specific to the context in which the data will be rendered (e.g., HTML, JavaScript, CSS) to provide precise protection against

## THE END