



INT6005CEM

BCSCUN

CS1

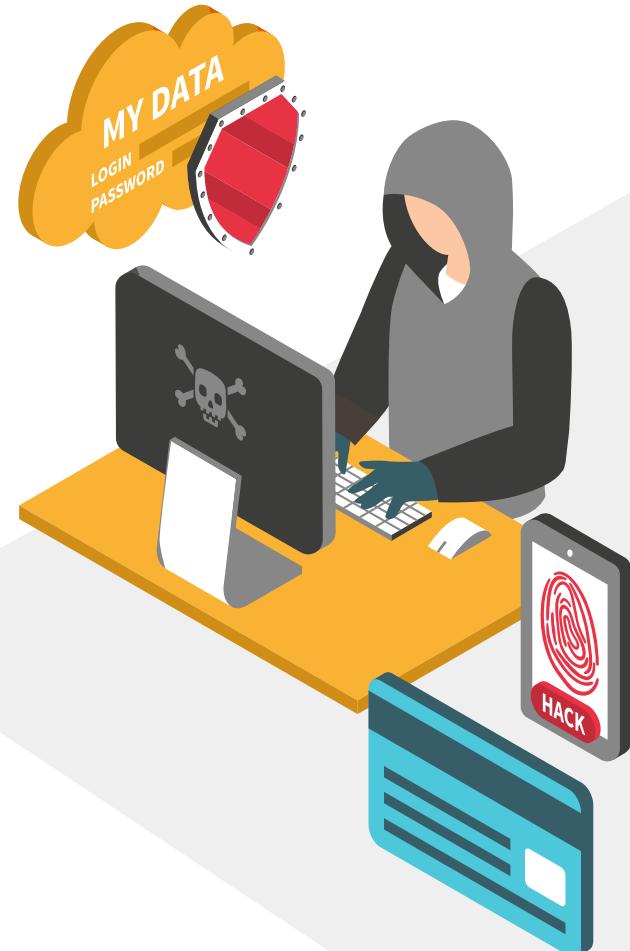
Lau En Sin (15731047)

Arief bin Abdul Latib (15730888)

Bryan Yeoh Seng Sheng (15730936)

Hsia Hao Ze (15730992)

Khor Cojean (15731014)



NECESSARY LINKS

GITHUB LINK

- Original System GitHub Link

<https://github.com/m-2199015/Online-Computer-Store.git>

- Enhanced System GitHub Link

https://github.com/Ensin2004/INT6005CEM_group_assignment_G23.git

USER MANUAL

- https://github.com/Ensin2004/INT6005CEM_group_assignment_G23/tree/main/User%20Manual

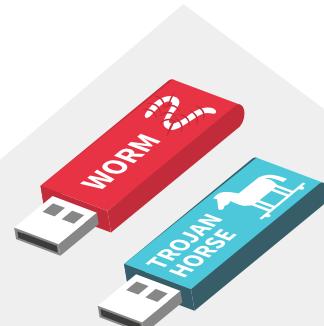




TABLE OF CONTENTS

01

SYSTEM OVERVIEW

02

SECURITY ANALYSIS

03

RISK ASSESSMENT

04

SECURITY RECOMMENDATION

05

SECURITY IMPLEMENTATION

06

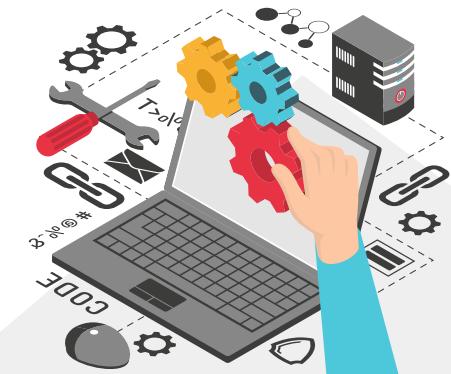
DISCUSSION & SUMMARY





01.

SYSTEM OVERVIEW



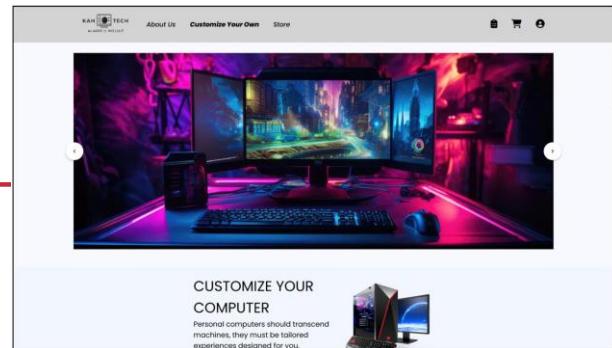
OUR SYSTEM - KAHTECH

KAHTECH

Online computer store platform for customers in to browse PC components, and place order.

Features

Explore accessories, view product details and customize PC builds



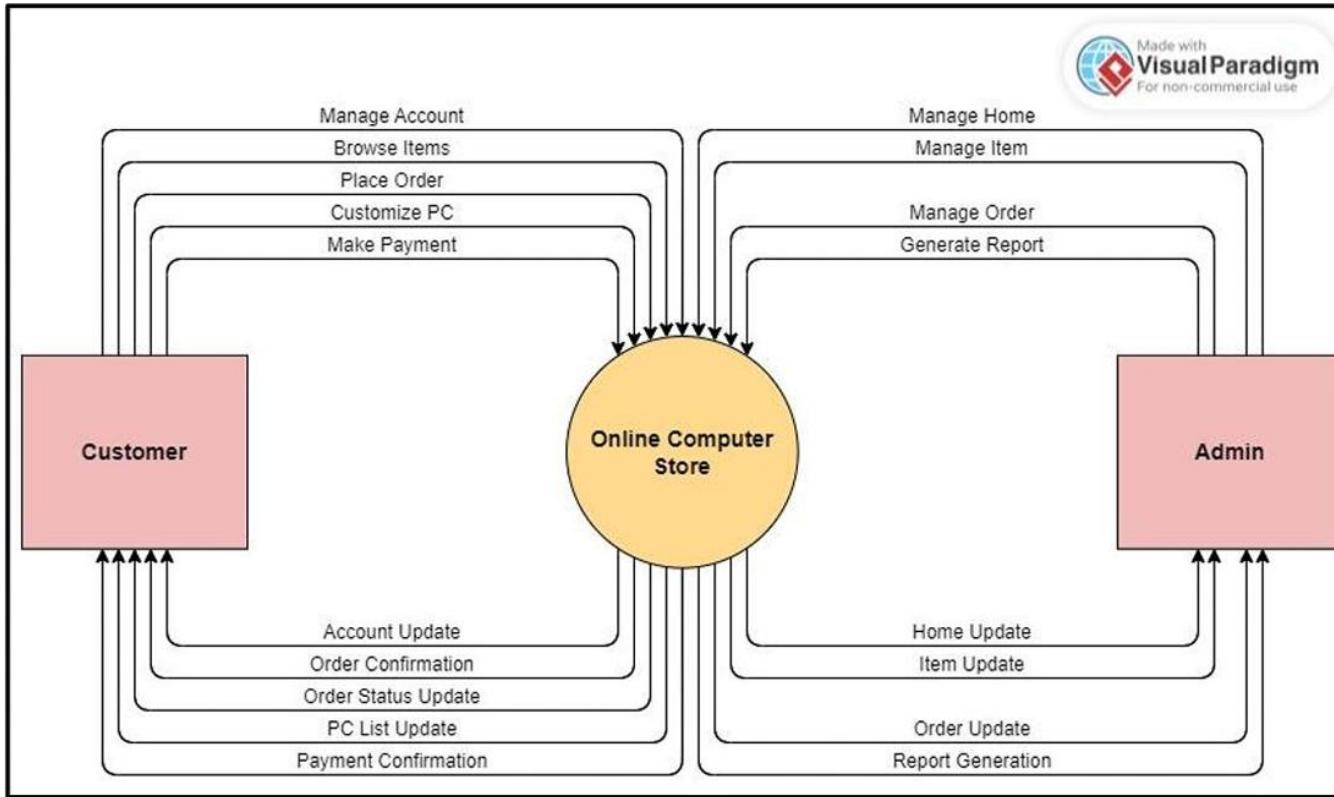
User roles

User and Admin

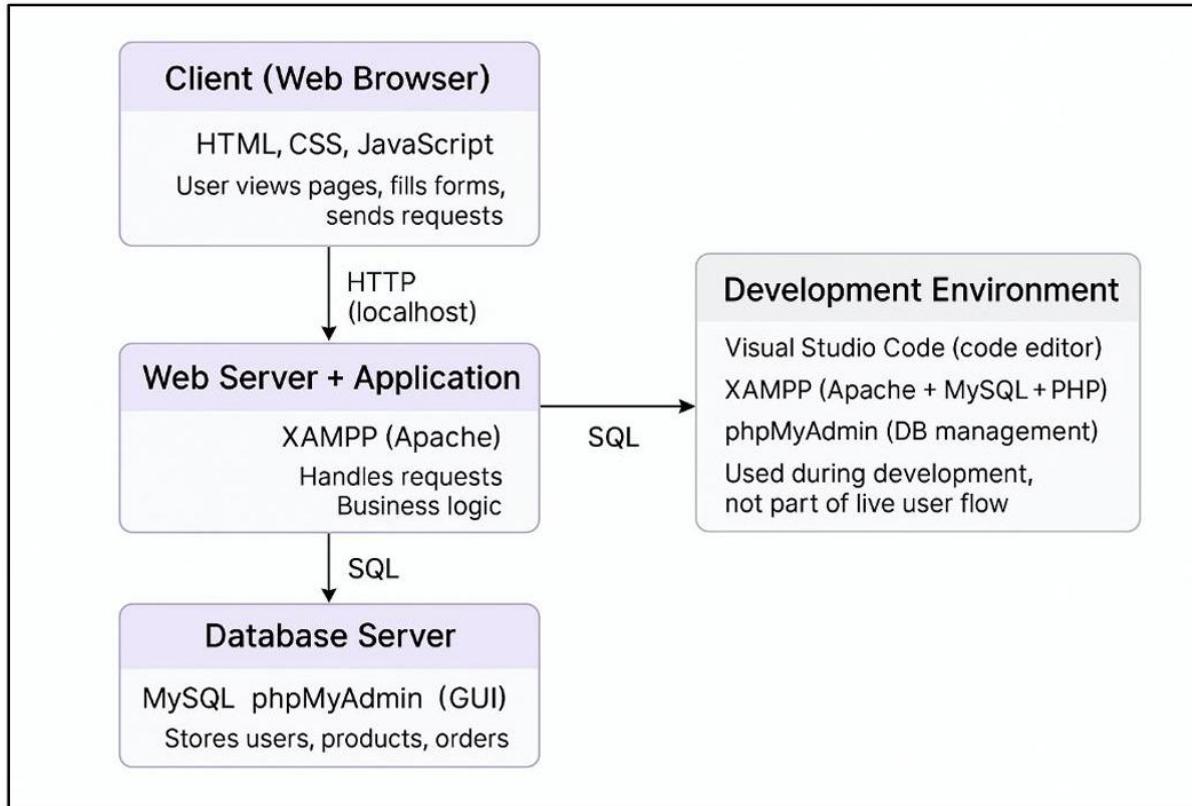
Ordering & Admin

Users can order with TNG e-Wallet or COD, Admins manage products, stock orders and reports

SYSTEM FUNCTIONALITIES



SYSTEM ARCHITECTURE





02.

SYSTEM ANALYSIS





SYSTEM ANALYSIS

01

Authentication

Lack of login rate limiting, super admin and forgot password features.

03

Email Account Handling

Lack of email verification and backup email implementation.



02

Password Security

Lack of strong password and password hashing implementation.

04

Data Privacy & User Education

Lack of data privacy policy and user education.



SYSTEM ANALYSIS

05

Session Management

Lack of session timeout, protection against session fixation and CSRF attacks.

07

Error Handling

Lack of page and system error handling.



06

Audit Logging

Lack of audit logging feature.

08

Input Validation & Cleaning

Lack of input validation, auto-trim, whitespace check and length input limit.



SYSTEM ANALYSIS

09

SQL Injection

Lack of protection against error-based, union-based and blind SQL injection.

11

Cryptography

Lack of protection against data-in-transit and data-at-rest.



10

Cross-Site Scripting (XSS)

Lack of protection against reflected, stored and DOM-based XSS.

12

Secure Cookies

Lack of secure cookies implementation.



SYSTEM ANALYSIS

13

Digital Certificate

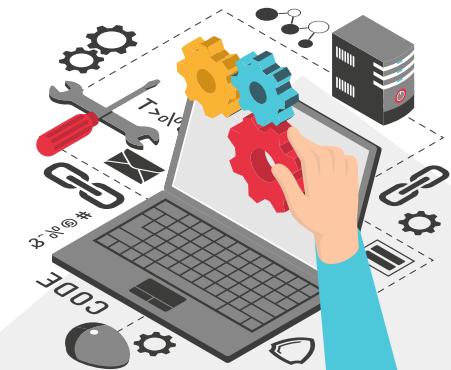
Lack of digital certificate deployment.





03.

RISK ASSESSMENT

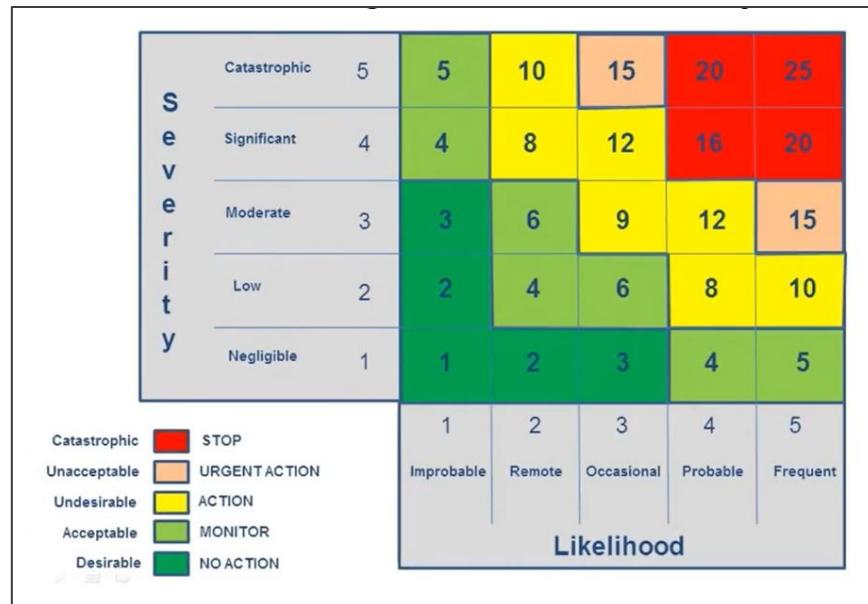




Risk Assessment

$$\text{Risk Score} = \text{Likelihood} \times \text{Severity}$$

- Likelihood: 1 (Improbable) to 5 (Frequent)
- Severity: 1 (Negligible) to 5 (Catastrophic)





Risk Assessment

Vulnerability	Likelihood (1-5)	Severity (1-5)	Risk Score/ Level (L×S)	Reason
Lack of <u>login</u> rate-limiting	5	5	25 (Catastrophic - STOP)	Unlimited brute-force attempts allow passwords to be guessed easily.
Missing super admin control	4	5	20 (Catastrophic - STOP)	A compromised manager will gain full system control, no principle of least privilege.
No <u>forgot-</u> password feature	3	3	9 (Action)	Users risk permanent lockout; more social-engineering opportunities.
Weak password rules	5	5	25 (Catastrophic - STOP)	Simple passwords are easily cracked by brute force.
No password hashing	5	5	25 (Catastrophic - STOP)	Plaintext passwords lead to instant account takeover.





Risk Assessment

No email verification	5	4	20 (Catastrophic - STOP)	Fake accounts, bots , spam and DDOS-style flooding.
No backup email	3	3	9 (Action)	Permanent lockout if primary email fails.
Missing privacy policy	3	3	9 (Action)	Unclear consent and possible compliance issues.
No session timeout	4	5	20 (Catastrophic - STOP)	Unattended sessions expose admin functions.
Potential session fixation	5	5	25 (Catastrophic - STOP)	Attacker can pre-set the victim's session ID.
No CSRF protection	5	5	25 (Catastrophic - STOP)	Forced actions possible without user intent.
No audit logging	3	3	9 (Action)	Hard to detect misuse or suspicious activity.





Risk Assessment

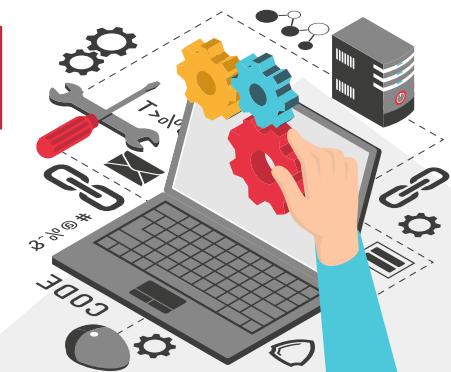
Insecure error messages	4	4	16 (Unacceptable - Urgent Action)	Server information leaks assist targeted attacks.
Missing input validation	5	5	25 (Catastrophic - STOP)	Opens the door to SQLi, XSS and malformed data.
SQL Injection	5	5	25 (Catastrophic - STOP)	Full database compromise is possible.
XSS (Reflected & Stored)	5	5	25 (Catastrophic - STOP)	Script execution leads to session theft and data manipulation.
Cryptographic failures (HTTP)	5	5	25 (Catastrophic - STOP)	Credentials travel in clear text; easily intercepted.
Cryptographic failures (plain DB)	5	5	25 (Catastrophic - STOP)	Full exposure of names, emails, phone, address.
Insecure cookies	5	5	25 (Catastrophic - STOP)	Session hijacking becomes trivial.
No SSL certificate	5	5	25 (Catastrophic - STOP)	MITM, sniffing and data tampering during transit.





04.

SECURITY RECOMMENDATION





Security Recommendation

Area	Recommendation
1. Authentication	<ul style="list-style-type: none">• Apply login rate limiting to slow down brute-force attempts.• Introduce a dedicated super-admin role to prevent privilege misuse.• Provide a secure forgot-password flow with OTP verification.
2. Password Security	<ul style="list-style-type: none">• Enforce strong password rules with minimum length and required character types.• Hash all passwords using Argon2id with high-cost parameters before storing.
3. Multi-Factor Authentication & Email Control	<ul style="list-style-type: none">• Verify user email during registration through a one-time code.• Add a backup email feature to support secure account recovery with MFA protection.
4. Data Privacy & User Awareness	<ul style="list-style-type: none">• Display clear privacy information on data usage.• Remind users about safe password handling and account hygiene.
5. Session Management & CSRF Protection	<ul style="list-style-type: none">• Implement automatic session timeout after inactivity.• Regenerate session IDs after login to prevent fixation.• Protect all sensitive forms with a server-generated CSRF token.
6. Logging & Monitoring	<ul style="list-style-type: none">• Record all admin actions and store before/after states for traceability.• Restrict log visibility to super-admins only.





Security Recommendation

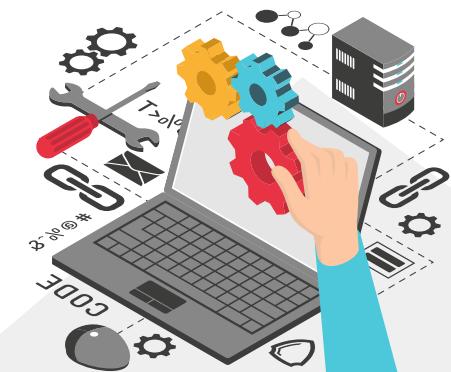
7. Error Handling	<ul style="list-style-type: none">• Use a unified error page for unexpected failures.• Remove sensitive details from all error outputs to avoid information leakage.
8. Input Validation	<ul style="list-style-type: none">• Validate field lengths and content for all user inputs.• Trim whitespace, reject invalid characters and give clear feedback for corrections.
9. SQL Injection Prevention	<ul style="list-style-type: none">• Sanitize incoming data and escape special characters where needed.• Use parameterized queries (prepare-bind-execute) for all database operations.
10. XSS Mitigation	<ul style="list-style-type: none">• Encode output in the correct context using <code>htmlspecialchars</code>, <code>urlencode</code> and safe link builders.• Ensure no untrusted input is reflected back without encoding.
11. Digital Certificate Deployment	<ul style="list-style-type: none">• Enable HTTPS with a valid SSL/TLS certificate.• Force HTTP→HTTPS redirection.• Enable HSTS for strict secure communication.
12. Cryptography	<ul style="list-style-type: none">• Protect all traffic with HTTPS/TLS 1.3 using a valid certificate.• Encrypt sensitive fields at rest with AES-256-GCM, using secure key handling.
13. Browser Cookie Security	<ul style="list-style-type: none">• Mark session cookies as Secure, HttpOnly and SameSite=Strict to reduce hijacking risks.





05.

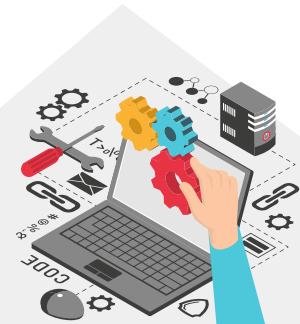
SECURITY IMPLEMENTATION





05.

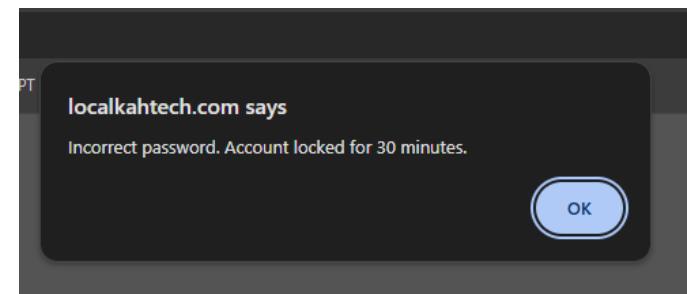
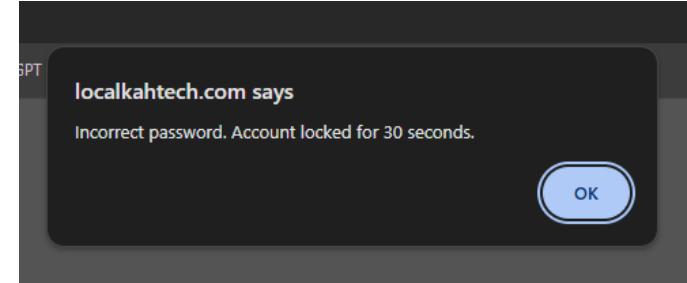
SECURITY IMPLEMENTATION (1. AUTHENTICATION)





Login Rate Limiting

```
39
40     $current_time = date("Y-m-d H:i:s");
41
42     // Check whether account is locked or not
43     if (!is_null($row['lock_until']) && $row['lock_until'] > $current_time) {
44         $remaining = strtotime($row['lock_until']) - time();
45         echo "<script> alert('Account is locked. Please try again after (" . $remaining . " seconds.');" . window.location.href='..../login.php'; </script>";
46         exit();
47     }
48
49     // Check whether password is correct or not
50     if (!password_verify($password, $row['pwd'])) {
51         $wrong_pwd_count = $row['wrong_pwd_count'] + 1;
52         $lock_until = null;
53         $lock_message = '';
54
55         // Account lock time
56         if ($wrong_pwd_count < 6) {
57             $lock_until = null;
58             $lock_message = "Incorrect password.";
59         } elseif ($wrong_pwd_count == 6) {
60             $lock_until = date("Y-m-d H:i:s", strtotime("+30 seconds"));
61             $lock_message = "Incorrect password. Account Locked for 30 seconds.";
62         } elseif ($wrong_pwd_count == 7) {
63             $lock_until = date("Y-m-d H:i:s", strtotime("+1 minute"));
64             $lock_message = "Incorrect password. Account Locked for 1 minute.";
65         } elseif ($wrong_pwd_count == 8) {
66             $lock_until = date("Y-m-d H:i:s", strtotime("+5 minutes"));
67             $lock_message = "Incorrect password. Account Locked for 5 minutes.";
68         } elseif ($wrong_pwd_count == 9) {
69             $lock_until = date("Y-m-d H:i:s", strtotime("+10 minutes"));
70             $lock_message = "Incorrect password. Account Locked for 10 minutes.";
71         } elseif ($wrong_pwd_count > 9) {
72             $lock_until = date("Y-m-d H:i:s", strtotime("+30 minutes"));
73             $lock_message = "Incorrect password. Account Locked for 30 minutes.";
74         }
75
76         // Update database
77         mysql_query(
78             $conn,
79                 "UPDATE users SET wrong_pwd_count = '$wrong_pwd_count', lock_until = " . ($lock_until ? "'$lock_until'" : "NULL") . " WHERE id = '" . $row['id'] . "'"
80         );
81
82         // Display messages
83         echo "<script> alert('$lock_message'); window.location.href='..../login.php'; </script>";
84
85     } else {
```



Protect the system from brute-force and automated password-guessing attacks.



Super Admin Role Setup

```
$_SESSION['ID'] = $row['id'];
$_SESSION['AdminName'] = $row['admin_name'];
$_SESSION['role'] = $row['role'];
$_SESSION['status'] = $row['account_status'];
```

```
<?php if (isset($_SESSION['role']) && $_SESSION['role'] === 'super_admin') { ?>
    <a class="menu_button" href="managers.php">
        <i class="fa-solid fa-users-gear"></i>
        <p class="menu_text">MANAGERS</p>
    </a>
</?php } ?>
```

```
4 if (isset($_GET['id']) && isset($_GET['action'])) {
5     $adminId = $_GET['id'];
6     $action = $_GET['action'];
7
8     if ($action === 'ban') {
9         $sql = "UPDATE admins SET account_status = 'banned' WHERE id = ?";
10    } elseif ($action === 'unban') {
11        $sql = "UPDATE admins SET account_status = 'active' WHERE id = ?";
```

```
if (isset($row['account_status']) && strtolower(string: $row['account_status']) === 'banned') {
    echo "<script>alert('Your account has been banned. Please contact the system administrator.');" . window.history.go(-1);</script>";
    exit();
}
```



Role Based Access Control, managers do not have access to “managers” page, only admins are able to access it and manage managers.



Forgot Password Mechanism

```
$checkEmail = mysqli_num_rows(result: mysqli_query(mysql: $conn, query: "SELECT email FROM users WHERE LOWER(email) = LOWER('$email') OR LOWER(second_email) = LOWER('$email')"));  
  
if ($checkEmail == 0) {  
    echo "<script>alert('Email not exists'); window.history.go(-1);</script>";  
    exit;  
  
$otp = rand(min: 10000, max: 99999);  
$otpExpiresAt = time() + 60;  
// Send OTP via email  
$to = $email;  
$from = "kahtechpng@gmail.com";  
$fromName = "KAHTECH";  
$message = $otp . " is your OTP.";  
$subject = "Secondary Email Verification";  
$header = 'From: ' . $fromName . ' <' . $from . '>';  
  
mail(to: $to, subject: $subject, message: $message, additional_headers: $header)
```

Welcome to

KAH TECH
WE BUILD PC NOT CRACK!

Email / Secondary Email:
aimarie0919@gmail.com

New Password :

Confirm Password

Enter OTP :

Send OTP Again (8s)

Verify OTP

Avoid permanent lock out when user forget the password, also prevent social engineering risk especially for those who store their login information on another platform or physically.





05.

SECURITY IMPLEMENTATION **(2. PASSWORD SECURITY)**





Strong Password Validation

```
script>
document.getElementById('newPassword').addEventListener('input', validatePassword);
document.getElementById('confirmPassword').addEventListener('input', validatePassword);

function validatePassword() {
    var password = document.getElementById('newPassword').value;
    var confirmPassword = document.getElementById('confirmPassword').value;
    var pwd_validation_container = document.getElementById('pwd_validation_container');
    var pwd_confirmation = document.getElementById('pwd_confirmation');
    var pwd_character = document.getElementById('pwd_character');
    var pwd_letter = document.getElementById('pwd_letter');
    var pwd_number = document.getElementById('pwd_number');
    var pwd_symbol = document.getElementById('pwd_symbol');
    var submit_btn = document.getElementById('submit_btn');

    const passwordRegex = /^(?=.*[a-zA-Z])(?=.*\d)(?=.*[$!@#%^&])[A-Za-z\d$!@#%^&]{8,20}$/;

    if (!passwordRegex.test(password) || confirmPassword != password) {
        submit_btn.disabled = true;
    }

    if (!passwordRegex.test(password)) {
        pwd_validation_container.style.display = "block";
        pwd_confirmation.style.display = "none";
    }

    if (password.length < 8 || password.length > 20) {
        pwd_character.style.display = "block";
    } else {
        pwd_character.style.display = "none";
    }

    if (!/[a-zA-Z]/.test(password)) {
        pwd_letter.style.display = "block";
    } else {
        pwd_letter.style.display = "none";
    }

    if (!/\d/.test(password)) {
        pwd_number.style.display = "block";
    } else {
        pwd_number.style.display = "none";
    }

    if (!/[!@#%^&]/.test(password)) {
        pwd_symbol.style.display = "block";
    } else {
        pwd_symbol.style.display = "none";
    }
}

else {
    pwd_validation_container.style.display = "none";
    pwd_confirmation.style.display = "block";
    submit_btn.disabled = false;
}
}


```



Name :

Email :

Phone Number :

Address :

New Password :

Password requirements:
* 8-20 characters
* at least one letter (A-Z)
* at least one number (0-9)
* at least one special characters (@\$!%*?&)
* no spaces allowed

Confirm Password

I confirm that I have read and agree to KAH TECH
 User Agreement and [Privacy & User Education Policy](#).

Sign up

Already have an account? [Log In](#).

Enforced strong password validation, requiring 8-20 characters, one letter, one number, one special character and no spaces allowed.





Password Hashing & Salting

```
$ARGON_OPTS = [  
    'memory_cost' => 131072, // 128 MB  
    'time_cost'     => 3,      // 3 iterations  
    'threads'       => 1  
];
```

```
'password = password_hash($_POST["newPassword"], PASSWORD_ARGON2ID, $ARGON_OPTS);
```

16	Kamen Rider	p22014743@student.newini.edu.my	019-8284731	Seri Melati, George Town	Kamen123@
17	Tan Mei Mei	p22014743@student.newini.edu.my	017-2374621	Air Itam Dam	Tanmeime123@
19	Junzo	junzobry2@gmail.com	0123456789	bryan 123, at jalan bryan	\$argon2id\$v=19\$m=131072,t=3,p=1\$8kxTbjjW7ktORjZEZE5kbw\$Hi/rZQBEULzlF+3Y8KpgczH5z0VVUMBRYvGEWM8dfEg

\$argon2id\$v=19\$m=131072,t=3,p=1\$8kxTbjjW7ktORjZEZE5kbw\$Hi/rZQBEULzlF+3Y8KpgczH5z0VVUMBRYvGEWM8dfEg			
	Meta Data	Salt	Hash



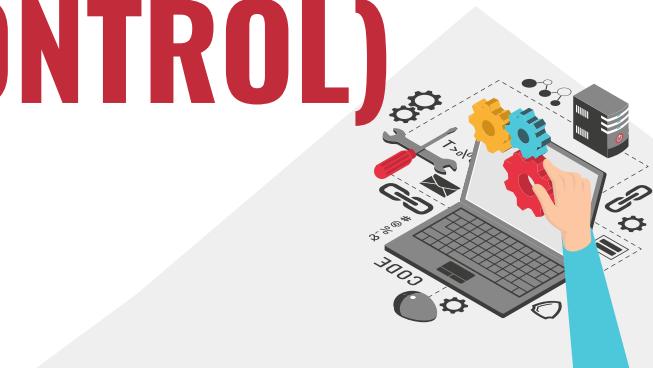
Converts passwords into irreversible hashes with unique salts instead of plaintext so stolen database data cannot reveal the original passwords



05.

SECURITY IMPLEMENTATION

(3. MFA & EMAIL CONTROL)





Email Verification

```
$checkResult = mysqli_num_rows(result: mysqli_query(mysql: $conn,
query: "SELECT user_name FROM users WHERE LOWER(user_name) = LOWER('$name');");
 
$checkEmail = mysqli_num_rows(result: mysqli_query(mysql: $conn,
query: "SELECT email FROM users WHERE LOWER(email) = LOWER('$email');");
 
$checkSecondaryEmail = mysqli_num_rows(result: mysqli_query(mysql: $conn,
query: "SELECT secondary_email FROM users WHERE LOWER(secondary_email) = LOWER('$email');"));


```

```
$otp = rand(min: 10000, max: 99999);
$otpExpiresAt = time() + 60;
// Send OTP via email
$to = $email;
$from = "kahtechpng@gmail.com";
$fromName = "KAHTECH";
$message = $otp . " is your OTP.";
$subject = "New Account Sign Up";
$header = 'From: ' . $fromName . ' <' . $from . '>';

mail(to: $to, subject: $subject, message: $message, additional_headers: $header)
```

The screenshot shows a registration or login form for 'KAH TECH'. The form fields include:

- Name: Arief Student
- Email: p22014743@student.newinti.edu.my
- Phone Number: 0123456789
- Address: Taman Seri Sarl
- New Password: (redacted)
- Confirm Password: (redacted)
- Enter OTP: (redacted)
- Send OTP Again (8s)
- Verify OTP (blue button)

Below the form, there is a note: "WE BUILT FC NOT CARST". At the bottom right, it says "Already have an account? Log in."

Username and email addresses that are used cannot be used again and OTP will be sent for verification so one email can only sign up for one time to prevent any spamming of bot accounts which may lead to DDOS attack.





Backup Email / Recovery with MFA

```
$checkEmail = mysqli_num_rows(result: mysqli_query(mysql: $conn, query: "SELECT email FROM users WHERE LOWER(email) = LOWER('$email') OR LOWER(secondary_email) = LOWER('$email');");
$query = mysqli_query(mysql: $conn, query: "SELECT * FROM users WHERE email = '$firstEmail'");
$row = mysqli_fetch_assoc(result: $query);

if (mysqli_num_rows(result: $query) == 0 || $row['pwd'] != $confirmPassword) {
    echo "<script> alert('Wrong password'); window.history.go(-1); </script>";
    exit;
}

if ($checkEmail != 0) {
    echo "<script>alert('Email already exists'); window.history.go(-1);</script>";
    exit;
}

$noAutoSend = isset($_POST['no_autosend']) || isset($_GET['no_autosend']);
$otp = $_POST['otp'] ?? $_GET['otp'] ?? null;
$otpExpiresAt = $_POST['otp_expires_at'] ?? $_GET['otp_expires_at'] ?? null;

$firstEmail = htmlspecialchars(string: $_POST['primaryEmail'] ?? $_GET['primaryEmail'] ?? '');
$email = htmlspecialchars(string: $_POST['secondaryEmail'] ?? $_GET['secondaryEmail'] ?? '');
$confirmPassword = htmlspecialchars(string: $_POST['confirmPassword'] ?? $_GET['confirmPassword'] ?? '');

// Send OTP via email
$to = $email;
$from = "kahtechpn@gmail.com";
$fromName = "KAHTECH";
$subject = "Secondary Email Verification";
$header = 'From: ' . $fromName . '<' . $from . '>';

if (($otp === null) && !$noAutoSend) || isset($_POST['resend'])) {
    $otp = rand(min: 10000, max: 99999);
    $otpExpiresAt = time() + 60; // seconds
    $message = $otp . " is your OTP.";
    @mail(to: $to, subject: $subject, message: $message, additional_headers: $header);
}
```

Name :

Email :

Secondary Email (Optional) :

Phone Number :

Address :

New Password :

Confirm Password :

Email :

Enter OTP :

Send OTP Again

If the main email has problems, the backup allows users to have a second way to prove ownership which prevent users from being permanently unavailable from the system.

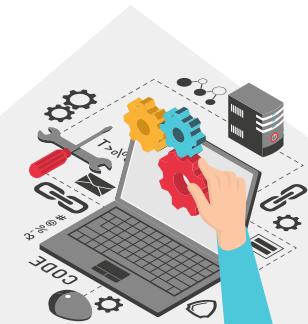




05.

SECURITY IMPLEMENTATION

(4. DATA PRIVACY & USER EDUCATION)





Data Privacy & User Education

```
<!-- === consent checkbox === -->


<label style="display:flex; gap:8px; align-items:flex-start; line-height:1.4;">
    <input type="checkbox" id="agree_terms" style="margin-top:3px;">
    <span>
      I confirm that I have read, consent to, and agree to KAH TECH
      <a href="user-agreement.php" target="_blank" rel="noopener">User Agreement</a>
      and
      <a href="privacy-education.php" target="_blank" rel="noopener">Privacy & User Education Policy</a>
    </span>
  </label>
  <p id="agree_error" style="display:none; color:#c8392b; margin-top:6px;">
    Please tick the checkbox to agree before signing up.
  </p>


```

KAH TECH
WE BUILT IT, NOT CRACKED!

Name :

Email :

Phone Number :

Address :

New Password :

Confirm Password :

I confirm that I have read, consent to, and agree to KAH TECH User Agreement and Privacy & User Education Policy.

▲ Please tick the checkbox to agree to the User Agreement and Privacy Policy.

Sign up

Already have an account? [Log in](#).

User agreement

User Agreement

This Agreement governs your access and use of the KAH TECH websites, applications, and related services. By clicking on accept or using our platforms, you agree to these terms.

Agree to:

- 1 Acceptable Use
- 2 Eligibility & Account
- 3 Acceptable Use
- 4 Intellectual Property
- 5 Privacy
- 6 Service Availability
- 7 Disclaimers & Limitation
- 8 Complaints & Feedback

1 Acceptance of Terms

By agreeing for an account, accessing, or using our service, you agree to be bound by this user agreement and our [Privacy & User Education Policy](#). If you do not agree, do not use the services.

2 Eligibility & Account

You must be at least 18 years old or have legal guardian consent to create an account.

- You are responsible for the confidentiality of your credentials and not disclose them under your account.
- Provide accurate information prior to when you change.

I'm over 18 years old and never receive password access different sites.

3 Acceptable Use

You agree not to misuse the service. Prohibited behavior include (without limitation):

- attempting unauthorized access, probing, or security testing without explicit permission;
- transferring resources, bandwidth or engaging in activities that distract or degrade service;
- attempting to damage or destroy another user's data or system, or violate their privacy or rights;

We may investigate violations and comply with law enforcement or regulatory requests.

4 Intellectual Property

All trademarks, logos, marks, designs, and content are the property of KAH TECH or its licensors and are protected by applicable law. You may not copy, modify, or distribute materials without prior written consent.

Privacy & user education

Privacy & User Education Policy

At KAH TECH, protecting your data and empowering you to stay safe online are top priorities. This page contains our privacy commitments with clear guidance for secure and responsible use.

Agree to:

- 1 Privacy Overview
- 2 Data We Collect
- 3 How We Use Data
- 4 Security Measures
- 5 Cookies & Analytics
- 6 User Feedback
- 7 Reporting Abuse
- 8 Complaints & Feedback

1 Privacy Overview

We collect personal information to provide and improve our service, do not sell or rent personal data, allow users or third party providers (hosting, analytics) they are bound by confidentiality and security obligations.

2 Data We Collect

- Account login name, email, phone number, address (provided during registration/app)
- Technical/device info: address, browser type, device identifiers, pages visited, installations (for security, diagnostics, and performance).

3 How We Use Data

- It collects/changes your account, deliver core features, and provide support.
- To measure service reliability, prevent fraudulence, and improve user experience.
- To send promotional material (news updates, service alerts). Marketing messages are sent only with consent and can be opted out at any time.

4 Security Measures

We apply reasonable technical and organizational measures, including encryption in transit, password hashing of user accounts, audit logs, and vulnerability monitoring. No method of communication is 100% secure, and we continuously improve our mitigation.

I'm aware of being unique password and enable 2FA/Bio metric systems. Avoid sharing passwords across different sites.

5 Cookies & Analytics

We use cookies to keep you signed in, remember preferences, and understand usage. You can control cookies in your browser settings. Blocking cookies may impact some functionality.

These controls ensure every user clearly understands the terms, privacy practices and safety guidelines before creating an account





05.

SECURITY IMPLEMENTATION

(5. SESSION MANAGEMENT & CSRF PROTECTION)

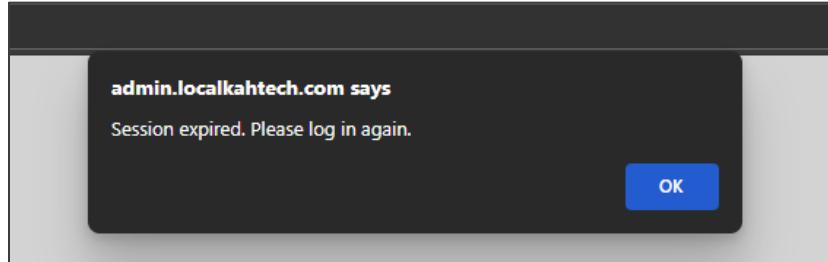




Session Timeout

```
JS sessionTimeout.js ✘ security.php
js > JS sessionTimeout.js > ...
1 // Timeout minutes
2 const timeoutMinutes = 5;
3
4 // Timeout seconds
5 const timeoutSeconds = timeoutMinutes * 60;
6
7 // Idle time counter in seconds
8 let idleTime = 0;
9
10 // Session flag
11 let sessionExpired = false;
12
13 // Reset idle time counter
14 function resetIdleTime() {
15     | idleTime = 0;
16 }
17
18 // Increment idle time every 10 seconds
19 const idleInterval = setInterval(() => {
20     if (sessionExpired) return;
21
22     idleTime += 10;
23
24     // Log out if idle time = timeout time
25     if (idleTime >= timeoutSeconds) {
26         sessionExpired = true;
27         window.location.href = "includes/logoutAccount.php?timeout=1";
28     }
29 }, 10000);
30
31 // Detect user activity and reset idle time counter
32 document.addEventListener("mousemove", resetIdleTime);
33 document.addEventListener("click", resetIdleTime);
34 document.addEventListener("scroll", resetIdleTime);
35 document.addEventListener("keydown", resetIdleTime);
```

```
16
17 // Session Timeout Check
18 $timeoutSeconds = 300;
19
20 if (isset($_SESSION['LastActivity']) && (time() - $_SESSION['LastActivity']) >= $timeoutSeconds) {
21     | echo "<script> window.location.href='includes/logoutAccount.php?timeout=1'; </script>";
22     | exit;
23 }
24
25 $_SESSION['LastActivity'] = time();
26
```



Protect the system from risks associated with unattended devices, shared computers and session misuse.



Session Regeneration

```
loginuser.php X logoutAccount.php  
includes > loginuser.php > ...  
88     mysqli_query(  
89         $conn,  
90         "UPDATE users SET wrong_pwd_count = 0, lock_until = NULL WHERE id = '{$row['id']}'"  
91     );  
  
    // Regenerate session ID (prevent session fixation)  
    session_regenerate_id(true);  
  
    // Update session  
    $_SESSION['ID'] = $row['id'];  
    $_SESSION['UserName'] = $row['user_name'];  
  
    // Display messages  
    echo "<script> alert('Log in successfully'); window.location.href='../../index.php'; </script>";  
100  
101
```

```
loginuser.php X logoutAccount.php X  
includes > logoutAccount.php > ...  
25     $params['httponly']  
26 );  
27 }  
  
// 3. Destroy the session on the server  
session_unset();  
session_destroy();  
  
// 4. Start new empty session with new session id  
session_start();  
session_regenerate_id(true);  
  
// 5. Redirect with alert  
echo "<script>alert('Log out successfully'); window.location.href='../../index.php';</script>";  
37  
38 exit;  
39
```

Before login

Cookie Name	Value	Decoded Value	Expires	Secure	HttpOnly	SameSite	Path	Domain	Host Only	Session
cfz_google...	%7B%22nzc_ga...	.cl...	/	2...	130	✓	✓	Lax		
cfz_reddit	%7B%22fZaD_re...	.cl...	/	2...	139	✓	✓	Lax		
kndctr_8A...	CY1NzQwOTcx...	.cl...	/	2...	137	✓	✓	Lax		
OptanonC...	isGpcEnabled=0...	.cl...	/	2...	391			Lax		
PHPSESSID	1lrqjke0cgpij4cs0ns...	lo...	/	S...	35	✓	✓	St...		
zaraz-cons...	{"Tuku":true,"aM...	.cl...	/	2...	38			St...		

Cookie Value Show URL-decoded
1lrqjke0cgpij4cs0nsd5afa5

After login

Cookie Name	Value	Decoded Value	Expires	Secure	HttpOnly	SameSite	Path	Domain	Host Only	Session
cfz_google...	%7B%22nzc_ga...	.cl...	/	2...	130	✓	✓	Lax		
cfz_reddit	%7B%22fZaD_re...	.cl...	/	2...	139	✓	✓	Lax		
kndctr_8A...	CY1NzQwOTcx...	.cl...	/	2...	137	✓	✓	Lax		
OptanonC...	isGpcEnabled=0...	.cl...	/	2...	391			Lax		
PHPSESSID	24chhpdm456atkr...	lo...	/	S...	35	✓	✓	St...		
zaraz-cons...	{"Tuku":true,"aM...	.cl...	/	2...	38			St...		

Cookie Value Show URL-decoded
24chhpdm456atkrupi4ea4g98

Attackers can no longer reuse or fix the session ID on the victim's device, mitigate the risk of session fixation attacks.





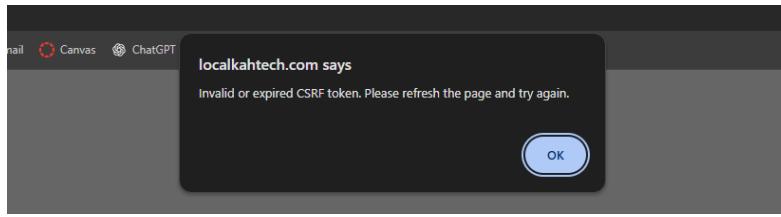
CSRF Token Implementation

```
  csrf.php ✘ updateUserAccount.php
includes? ✓ csrf.php? ...
3 // Start session if not started
4 if (session_status() === PHP_SESSION_NONE) {
5     session_start();
6 }
7
8 // Create new CSRF token if not set or expired
9 function createCSRFToken() {
10
11     if (!isset($_SESSION['CSRFToken']) || !isset($_SESSION['CSRFTokenExpiry']) || time() > $_SESSION['CSRFTokenExpiry']) {
12         $_SESSION['CSRFToken'] = bin2hex(random_bytes(32));
13         $_SESSION['CSRFTokenExpiry'] = time() + 1800;
14     }
15
16     return $_SESSION['CSRFToken'];
17 }
18
19 // Check CSRF token
20 function checkCSRFToken($token) {
21
22     // Ensure all variables are set
23     if (!isset($_SESSION['CSRFToken']) || !isset($_SESSION['CSRFTokenExpiry']) || !isset($token)) {
24         return false;
25     }
26
27     // Check CSRF token expiry
28     if (time() > $_SESSION['CSRFTokenExpiry']) {
29         unset($_SESSION['CSRFToken']);
30         unset($_SESSION['CSRFTokenExpiry']);
31         return false;
32     }
33
34     // Check CSRF token validity
35     if (hash_equals($_SESSION['CSRFToken'], $token)) {
36         unset($_SESSION['CSRFToken']);
37         unset($_SESSION['CSRFTokenExpiry']);
38         return true;
39     } else {
40         return false;
41     }
42 }
43
44 // Create hidden input for CSRF token
45 function createCSRFInput() {
46     $token = createCSRFToken();
47     echo '<input type="hidden" name="csrfToken" value="' . htmlspecialchars($token) . '">';
48 }
```

```
36     <!-- Biggest box in body to set background -->
37     <div class="editAccBox">
38         <div class="signUpContent">
39             <form class="accBox" action="includes/updateUserAccount.php" method="post" enctype="multipart/form-data">
40                 <?php createCSRFInput(); ?>
41                 <div class="signatureLogo">
42                     <img alt="Signature logo" data-bbox="106 78 890 250" />
43             </div>
44         </div>
45     </div>
```

```
  curl.php    updateUserAccount.php x
includes > updatedUserAccount.php ...
```

10 session_start();
11 require_once "dbh.inc.php";
12 require_once "csrf.php";
13
14 > \$ARGON_OPTS = [...]
15
16];
17
18 if (\$_SERVER["REQUEST_METHOD"] == "POST") {
19
20
21 // Check CSRF token
22 if (!isset(\$_POST['csrfToken']) || !checkCSRFToken(\$_POST['csrfToken'])) {
23 die("<script> alert('Invalid or expired CSRF token. Please refresh the page and try again.');" . window.history.go(-1) . "</script>");
24 }
25
26
27 // Collect form data
28 \$name = htmlspecialchars(trim(\$_POST['newUsername']));
29 \$email = htmlspecialchars(trim(\$_POST['newEmail']));
30 \$phone = htmlspecialchars(trim(\$_POST['newPhone']));



CSRF token prevents request from unauthorized websites to be executed.



05.

SECURITY IMPLEMENTATION (6. LOGGING & MONITORING)





Logging & Monitoring

Admin

Account

Audit Logs

Actor (Name ID) Action Outcome Entity Type Entity ID Export CSV

e.g. #2 e.g. admin_update All e.g. items e.g. 55 Reset Apply

From (date) To (date) Search summary Per Page mm/dd/yyyy mm/dd/yyyy contains... 20

Time	Actor	Action	Entity	Summary	Outcome	IP	View
2025-11-14 19:25:49	#1 super_admin	login_success	-	Admin logged in	success	1.1.1.1	View
2025-11-14 19:25:26	#1	login_failure	-	Admin login failed: name=admin, email=envin2004@gmail.com	failure	1.1.1.1	View
2025-11-14 19:25:08	#1 super_admin	logout	admins #2	Admin logged out	success	1.1.1.1	View
2025-11-14 18:44:53	#1 super_admin	login_success	-	Admin logged in	success	1.1.1.1	View
2025-11-14 18:44:34	#1 super_admin	login_failure	-	Wrong password for admin #1	failure	1.1.1.1	View
2025-11-14 18:43:55	#1 super_admin	logout	admins #2	Admin logged out	success	1.1.1.1	View
2025-11-14 18:24:15	#1 super_admin	item_soft_delete	items #3	Soft-deleted item #3	success	1.1.1.1	View

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		No	None		AUTO_INCREMENT	Change Drop More	
2	actor_admin_id	int(11)		Yes	NULL			Change Drop More	
3	actor_role	enum('manager','super_admin')	utf8mb4_general_ci	Yes	NULL			Change Drop More	
4	action	varchar(64)	utf8mb4_general_ci	No	None			Change Drop More	
5	entity_type	varchar(64)	utf8mb4_general_ci	Yes	NULL			Change Drop More	
6	entity_id	int(11)		Yes	NULL			Change Drop More	
7	summary	varchar(255)	utf8mb4_general_ci	No	None			Change Drop More	
8	before_json	longtext	utf8mb4_bin	Yes	NULL			Change Drop More	
9	after_json	longtext	utf8mb4_bin	Yes	NULL			Change Drop More	
10	outcome	enum('success','failure')	utf8mb4_general_ci	No	success			Change Drop More	
11	ip_address	varchar(45)	utf8mb4_general_ci	Yes	NULL			Change Drop More	
12	user_agent	varchar(255)	utf8mb4_general_ci	Yes	NULL			Change Drop More	
13	created_at	datetime		No	current_timestamp()			Change Drop More	

Audit Entry

Time: 2025-11-14 18:24:13
Actor: #1 (super_admin)
Action: item_soft_delete | Entity: items #3
Outcome: success | IP: 1.1.1.1
UA: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
Summary: Soft-deleted item #3

Before

```
{
  "id": "3",
  "item_name": "Razer Basilisk V3 Customizable Wired Chroma RGB Gaming Mouse",
  "item_status": "Active"
}
```

After

```
{
  "id": "3",
  "item_name": "Razer Basilisk V3 Customizable Wired Chroma RGB Gaming Mouse",
  "item_status": "Deleted"
}
```



Strengthens accountability, supports incident investigation and ensures all critical actions are fully traceable.



05.

SECURITY IMPLEMENTATION (7. ERROR HANDLING)





Unified Error Page

```
<VirtualHost *:443>
    ServerName www.localkahtech.com
    ServerAlias localkahtech.com
    DocumentRoot "C:/xampp/htdocs/INT6005CEM_group_assignment_G23/Online Computer Store"

    ServerSignature Off
    ErrorDocument 404 /errors/404.php
    ErrorDocument 500 /errors/500.php

    Alias /Image "C:/xampp/htdocs/INT6005CEM_group_assignment_G23/Image"

    <Directory "C:/xampp/htdocs/INT6005CEM_group_assignment_G23/Image">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile "conf/ssl.crt/server.crt"
    SSLCertificateKeyFile "conf/ssl.key/server.key"

    Header always set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

```
function handleErrorAndExit($message = 'Unexpected error during sign up.') {
    error_log('[SIGNUP ERROR] ' . $message);
    http_response_code(500);
    header("Location: ../errors/500.php");
    exit;
}
```

KAH TECH
WE BUILD FOR THE FUTURE!

404

Page not found

The page you're looking for doesn't exist, has been moved, or the link is incorrect.

[Back to Home](#)

KAH TECH
WE BUILD FOR THE FUTURE!

500

Something went wrong

An unexpected error occurred on our side. Please try again in a moment.

[Back to Home](#)



Provides a single safe fallback screen so unexpected failures show controlled messages without exposing system details.



Suppress Sensitive Details

```
display_errors=Off  
  
; The display of errors which occur during PHP's startup sequence are handled  
; separately from display_errors. We strongly recommend you set this to 'off'  
; for production servers to avoid leaking configuration details.  
; Default Value: On  
; Development Value: On  
; Production Value: Off  
; https://php.net/display-startup-errors  
display_startup_errors=Off  
  
; Besides displaying errors, PHP can also log errors to locations such as a  
; server-specific Log, STDERR, or a location specified by the error_log  
; directive found below. While errors should not be displayed on productions  
; servers they should still be monitored and Logging is a great way to do that.  
; Default Value: Off  
; Development Value: On  
; Production Value: On  
; https://php.net/Log-errors  
log_errors=On
```

```
24999 59 2025] [php:warn] [pid 51472:tid 1936] [client ::1:2988] PHP Warning: Undefined variable $img3_file_name in C:\\xampp\\htdocs\\Online-Computer-Store-main\\Admin Online Compu  
25000 59 2025] [php:warn] [pid 51472:tid 1936] [client ::1:2988] PHP Warning: Undefined variable $img4_file_name in C:\\xampp\\htdocs\\Online-Computer-Store-main\\Admin Online Compu  
25001 59 2025] [php:warn] [pid 51472:tid 1936] [client ::1:2988] PHP Warning: Undefined variable $img5_file_name in C:\\xampp\\htdocs\\Online-Computer-Store-main\\Admin Online Compu  
25002 59 2025] [php:error] [pid 51472:tid 1936] [client ::1:2988] PHP Fatal error: Uncaught mysqli_sql_exception: Unknown column 'abc' in 'field list' in C:\\xampp\\htdocs\\Online-Computer-Store-main\\Admin Online Compu  
25003 22 2025] [php:error] [pid 51472:tid 1944] [client 127.0.0.1:54030] script 'C:/xampp/htdocs/INT6005CEM_group_assignment_G23/Online Computer Store/hello.php' not found or unable  
25004 38 2025] [php:error] [pid 51472:tid 1944] [client 127.0.0.1:54030] script 'C:/xampp/htdocs/INT6005CEM_group_assignment_G23/Online Computer Store/hello.php' not found or unable  
25005 21 2025] [php:notice] [pid 51472:tid 1920] [client 127.0.0.1:31054] [LOGIN ERROR] Table 'computer_store.bryan' doesn't exist, referer: https://localkahtech.com/login.php  
25006
```



Hides internal system information so errors reveal only safe messages, preventing attackers from learning anything useful for exploitation



05.

SECURITY IMPLEMENTATION

(8. INPUT VALIDATION & CLEANING)





Input Validation

```
// ----- Name validation -----
function validateName() {
  const name = nameInput.value.trim();
  const error = document.getElementById('nameError');

  // Letters, spaces, apostrophes, dots, hyphens; length 2-50
  const nameRegex = /^[A-Za-z\s'.-]{2,50}$/;

  if (name.length === 0) {
    error.style.display = "none";
    isNameValid = false;
  } else if (nameRegex.test(name)) {
    error.style.display = "none";
    isNameValid = true;
  } else {
    error.style.display = "block";
    isNameValid = false;
  }

  updateSubmitButton();
}

// ----- Email validation -----
function validateEmail() {
  const error = document.getElementById('emailError');
  const email = emailInput.value.trim();

  if (email.length === 0) {
    error.style.display = "none";
    isEmailValid = false;
  } else if (emailInput.validity.valid) {
    error.style.display = "none";
    isEmailValid = true;
  } else {
    error.style.display = "block";
    isEmailValid = false;
  }

  updateSubmitButton();
}
```

```
// ----- Malaysian phone validation -----
function validatePhone() {
  const phone = phoneInput.value.trim();
  const error = document.getElementById('phoneError');

  // Simple Malaysian format:
  // - starts with 0
  // - total 10 or 11 digits
  const phoneRegex = /^0\d{8,9}$/;

  if (phone.length === 0) {
    error.style.display = "none";
    isPhoneValid = false;
  } else if (phoneRegex.test(phone)) {
    error.style.display = "none";
    isPhoneValid = true;
  } else {
    error.style.display = "block";
    isPhoneValid = false;
  }

  updateSubmitButton();
}
```



Please enter a valid name (letters only).

Name : Bryan12

Please enter a valid email address.

Email : bryanyeoh68@gmail.com

Please enter a valid Malaysian phone number (starting with 0).

Phone Number : 010550473a

New Password :

Confirm Password :

I confirm that I have read and agree to KAH TECH User Agreement and Privacy & User Education Policy.

User Agreement and Privacy & User Education Policy.

[Sign up](#)

Already have an account? [Log in](#).



Input validation checks whether submitted data follows the required rules so only safe, expected values enter the system, blocking harmful or invalid inputs.



Trim & Whitespace Checks

```
$name = htmlspecialchars(trim($_POST["newUsername"]));  
$email = htmlspecialchars(trim($_POST["newEmail"]));  
$phone = htmlspecialchars(trim($_POST["newPhone"]));  
$address = htmlspecialchars(trim($_POST["newAddress"]));  
$password = password_hash($_POST["newPassword"], PASSWORD_ARGON2ID, $ARGON_OPTS);
```

A screenshot of a web form for user registration. The form fields include:

- Name: Bryan
- Email: bryanyeh681@gmail.com
- Phone Number: 0105504731
- Address: haha hahahaha
- New Password: (obscured)
- Confirm Password: (obscured)

At the bottom, there is a checkbox labeled "I confirm that I have read and agree to KAH TECH User Agreement and Privacy & User Education Policy." followed by a "Sign up" button and a link "Already have an account? Log in."

19 Junzo	junzobryn28@gmail.com	0123456789	bryan 123, at jalan bryan	\$argon2id\$v=19\$m=131072,t=3,p=1\$NkxTbjJWtktORjZEZE...
20 Bryan	bryanyeh681@gmail.com	0105504731	haha hahahaha	\$argon2id\$v=19\$m=131072,t=3,p=1\$ZEV1T1ZCuktVQ1Q4aD...

 Trim and whitespace checks remove extra spaces and hidden characters so the system processes clean, accurate input and blocks sneaky manipulation.



Input Limiting

```
<div class="signUpInfo">
  <label for="Username">Name :</label>
  <input required type="text" id="Username" name="newUsername" maxlength="50" placeholder="Enter Username">
  <p class="limit-warning" id="nameLimit">Character limit reached (50)</p>
  <p class="validation-error" id="nameError">Please enter a valid name (letters only).</p>

  <label for="Email">Email :</label>
  <input required type="email" id="Email" name="newEmail" maxlength="100" placeholder="Enter Email">
  <p class="limit-warning" id="emailLimit">Character limit reached (100)</p>
  <p class="validation-error" id="emailError">Please enter a valid email @address.</p>

  <label for="Phone">Phone Number :</label>
  <input required type="text" id="Phone" name="newPhone" maxlength="11" placeholder="Enter Phone Number">
  <p class="limit-warning" id="phoneLimit">Character limit reached (11)</p>
  <p class="validation-error" id="phoneError">Please enter a valid Malaysian phone number (starting with 01).</p>

  <label for="Address">Address :</label>
  <input required type="text" id="Address" name="newAddress" maxlength="200" placeholder="Enter Address">
  <p class="limit-warning" id="addressLimit">Character limit reached (200)</p>

  <label for="newPassword">New Password :</label>
  <div class="pwd-wrapper">
    <input required type="password" id="newPassword" name="newPassword" maxlength="20" placeholder="Enter New Password">
    <i class="fa-solid fa-eye-slash toggle-eye" onclick="togglePassword('newPassword', this)"></i>
  </div>
  <p class="limit-warning" id="newPwdLimit">Character limit reached (20)</p>
  <div class="pwd_validation_container" id="pwd_validation_container">
    <p>Password requirements:</p>
    <p class="pwd_validation" id="pwd_character">* at least 8-20 characters</p>
    <p class="pwd_validation" id="pwd_letter">* at least one character (A-Z)</p>
    <p class="pwd_validation" id="pwd_number">* at least one number (0-9)</p>
    <p class="pwd_validation" id="pwd_symbol">* at least one special characters ($@!%?&)</p>
  </div>
  <p class="pwd_validation" id="pwd_space">* no spaces allowed</p>
  </div>

  <label for="confirmPassword">Confirm Password:</label>
  <div class="pwd-wrapper">
    <input required type="password" id="confirmPassword" name="confirmPassword" maxlength="20" placeholder="Confirm Password">
    <i class="fa-solid fa-eye-slash toggle-eye" onclick="togglePassword('confirmPassword', this)"></i>
  </div>
  <p class="limit-warning" id="confirmPwdLimit">Character limit reached (20)</p>
  <p class="pwd_confirmation" id="pwd_confirmation">Password not match</p>
</div>
```

```
// Field Limit Warning
function setupLimitWarning(inputId, warningId, max) {
  const input = document.getElementById(inputId);
  const warning = document.getElementById(warningId);

  input.addEventListener('input', () => {
    if (input.value.length === max) {
      warning.style.display = "block";
    } else {
      warning.style.display = "none";
    }
  });
}

// Initialize limit warnings for all fields
setupLimitWarning('Username', 'nameLimit', 50);
setupLimitWarning('Email', 'emailLimit', 100);
setupLimitWarning('Phone', 'phoneLimit', 11);
setupLimitWarning('Address', 'addressLimit', 200);
setupLimitWarning('newPassword', 'newPwdLimit', 20);
setupLimitWarning('confirmPassword', 'confirmPwdLimit', 20);

// ----- Attach validation listeners -----
nameInput.addEventListener('input', validateName);
emailInput.addEventListener('input', validateEmail);
phoneInput.addEventListener('input', validatePhone);
```

```
function validatePrice(input) {
  const warning = document.getElementById("priceWarning");
  if (parseFloat(input.value) > parseFloat(input.max)) {
    input.value = input.max;
    warning.style.display = "block";
  } else {
    warning.style.display = "none";
  }
}

function validateStock(input) {
  const warning = document.getElementById("stockWarning");
  if (parseInt(input.value) > parseInt(input.max)) {
    input.value = input.max;
    warning.style.display = "block";
  } else {
    warning.style.display = "none";
  }
}

// Field Limit Warning
function setupLimitWarning(inputId, warningId, max) {
  const input = document.getElementById(inputId);
  const warning = document.getElementById(warningId);

  input.addEventListener('input', () => {
    if (input.value.length === max) {
      warning.style.display = "block";
    } else {
      warning.style.display = "none";
    }
  });
}

// Initialize limit warnings for all fields
setupLimitWarning('name', 'nameLimit', 255);
setupLimitWarning('description', 'descriptionLimit', 999);
```

Input limiting restricts data size and format so only safe, valid inputs are accepted, blocking malicious or oversized submissions.



Input Limiting



User Sign Up Form Limit

Admin Add Item Form Limit



Admin

Item Name : INTEL CORE I7 12700hahsahdjsahdjhahshdashdjhshjoahdjhshaldhoshdjhshdhshdjhshoajhdhsahdjhksa
Character limit reached (295)

Description : i2700hahsahdjsahdjhahshdashdjhshjoahdjhshaldhoshdjhshdhshdjhshoajhdhsahdjhksahjkdhjkah
Character limit reached (995)

Category : -- Select Category --

Price (RM) : 999999999
Maximum value is 999999999

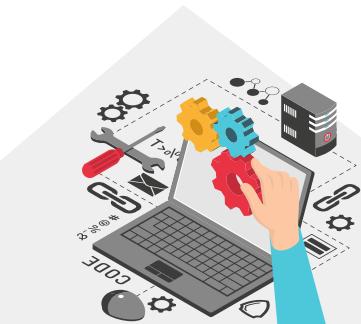
Stock : 9999
Maximum value is 9999

[ADD ITEM](#)



05.

SECURITY IMPLEMENTATION (9. SQL INJECTION PREVENTION)





SQL Prevention

Input Sanitization

```
/*
 * Basic sanitization only (no validation here):
 * - trim
 * - strip HTML tags
 * - remove special symbols (keeps letters/numbers/space/-/_/.,)
 */
3 references
function sanitize_basic(?string $s): string {
    if ($s === null) return '';
    $s = trim(string: strip_tags(string: $s));
    // allow: a-z, A-Z, 0-9, space, dash, underscore, dot, comma
    $s = preg_replace(pattern: '/[^a-zA-Z0-9 \-\_\.\,]/u', replacement: ' ', subject: $s);
    // collapse multiple spaces
    $s = preg_replace(pattern: '/\s+/u', replacement: ' ', subject: $s);
    return $s;
}
```

```
// Sanitize inputs
$category_raw = $_GET['category'] ?? null;
$search_raw   = $_GET['search'] ?? null;

$category = $category_raw !== null ? sanitize_basic(s: $category_raw) : null;
$search   = $search_raw  !== null ? sanitize_basic(s: $search_raw)  : null;
```

Together, these layered controls ensure all user inputs are safely handled and malicious SQL commands are fully prevented from reaching or manipulating the database



Escaping Special Characters

```
// utf8mb4 for safety
$conn->set_charset(charset: 'utf8mb4');

/
$like_esc = $conn->real_escape_string(string: "%{$search}%");
$sql_legacy = "
SELECT items.id, items.item_name, items.price, items.stock_qty, items.description,
       items.image1, items.image2, items.image3, items.image4, items.image5,
       categories.category_name
FROM items
LEFT JOIN categories ON items.category_id = categories.id
WHERE item_status = 'Active'
      AND (
              LOWER(items.item_name)          LIKE LOWER('{$like_esc}')
          OR LOWER(items.description)      LIKE LOWER('{$like_esc}')
          OR LOWER(categories.category_name) LIKE LOWER('{$like_esc}')")
```

Parameterized Queries

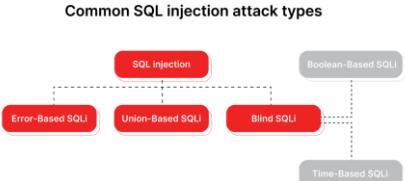
```
$like = "%{$search}%";
$sql = "
SELECT items.id, items.item_name, items.price, items.stock_qty, items.description,
       items.image1, items.image2, items.image3, items.image4, items.image5,
       categories.category_name
FROM items
LEFT JOIN categories ON items.category_id = categories.id
WHERE item_status = 'Active'
      AND (
              LOWER(items.item_name)          LIKE LOWER(?)
          OR LOWER(items.description)      LIKE LOWER(?)
          OR LOWER(categories.category_name) LIKE LOWER(?))
      ";

$stmt = $conn->prepare(query: $sql);

if ($stmt) {
    $stmt->bind_param(types: "sss", var: $like, vars: $like); // escaping not needed for prepared statements
    $stmt->execute();
    $itemResult = $stmt->get_result();
    $stmt->close();
```



SQL Prevention



Error based SQL Injection


KAH TECH
 WE BUILD THE BEST Laptops

Admin


Account

X

Q

ALL
MOUSE
KEYBOARD
MONITOR
GRAPHIC CARD
CPU
MEMORY
STORAGE
MOTHERBOARD

Results for ` AND EXTRACTVALUE(1, CONCAT(0x7e,(SELECT DATABASE()),0x7e)) --` - 0 items

No results for ` AND EXTRACTVALUE(1, CONCAT(0x7e,(SELECT DATABASE()),0x7e)) --`

Union based SQL Injection

Blind SQLi Injection (Boolean)

The screenshot shows a search interface with the following elements:

- Logo and Header:** "KAH TECH" logo with the tagline "WE BUILD IT, WE DESIGN IT".
- Title:** "Admin" displayed prominently.
- Search Bar:** A search bar containing the query "'OR' || --".
- Search Results:** Below the search bar, there is a row of blue buttons representing product categories: ALL, MOUSE, KEYBOARD, MONITOR, GRAPHIC CARD, CPU, MEMORY, STORAGE, and MOTHERBOARD.
- Text:** "Results for 'OR' || --" followed by "0 items".
- Text:** "No results for ' OR' || --'

Blind SQLi Injection (Time)



KAH TECH
WE BUILD IT, WE TEST IT

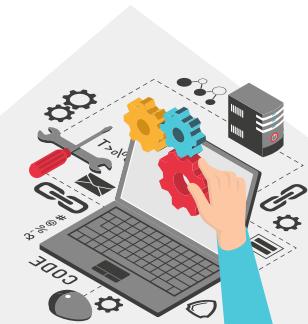
Admin

Account



05.

SECURITY IMPLEMENTATION (10. XSS MITIGATION)





XSS Mitigation

```
/**  
 * Output-encoding helpers (XSS mitigation by context)  
 */  
  
10 references  
function e(string $s): string {           // HTML text/attribute  
    return htmlspecialchars(string: $s ?? '', flags: ENT_QUOTES, encoding: 'UTF-8');  
}  
  
2 references  
function q(string $s): string {           // URL query parameter  
    return urlencode(string: $s);  
}  
  
1 reference  
function qp(string $s): string {          // URL path segment  
    return rawurlencode(string: $s);  
}
```

```
$catHref = "store.php?category=" . q(s: $catName);  
$cls = $isSelected ? 'category_button selected' : 'category_button';  
echo '<a class="'. $cls .'" href="'.$catHref.'>' .  
    strtoupper(string: e(s: $catName)) . '</a>';
```

```
<!-- search bar -->  
<form class="search_bar" action="store.php" method="get">  
    <div class="search_box">  
        <input  
            type="text"  
            name="search"  
            placeholder="Search"  
            value="<?php echo isset($_GET['search']) ? e(s: $_GET['search']) : ''; ?>"  
        >  
        <button type="submit"><i class="fa-solid fa-magnifying-glass"></i></button>  
    </div>  
</form>
```

```
$id    = (int)$row['id']; // numeric only  
$name  = e(s: $row['item_name']);  
$desc  = e(s: $row['description']);  
$price = e(s: $row['price']);  
// guard filename, then URL-encode for path usage  
$img1f = basename(path: $row['image1'] ?? '');  
$imgSrc = "../Image/" . qp(s: $img1f);  
$stock = (int)$row['stock_qty'];  
$detailsHref = "itemDetails.php?item={$id}";
```



Ensure that any malicious input is safely neutralised and displayed as harmless text, preventing scripts from ever running in the browser

XSS Mitigation



Script Tag

The screenshot shows a browser window with the URL `http://localhost:10205/SCIM_security_group_assignment/G23/Online Computer Store/Admin/2020/online%20Computer%20store/store.php?search=><script>alert(%21)%21</script>`. The page title is "Admin e Account". The developer tools Network tab is open, showing a POST request to `store.php?search=><script>alert(%21)%21</script>`. The response body contains the following XSS payload:

```
<div class="store_head">
    <!-- search bar -->
    <form action="store.php" method="get">
        <div class="search_bar">
            <input type="text"
                name="search"
                placeholder="Search"
                value="<script>alert(%21)%21</script>">
            <button type="submit"><i>fa-solid fa-magnifying-glass</i></button>
        </div>
        <!-- category bar -->
        <div class="category_bar">
            <a class="category_button selected" href="store.php?category=category_button" href="store.php?category=category_button">
                <i>fa-brands fa-google-wallet</i>
            </a>
            <a class="category_button" href="store.php?category=category_button" href="store.php?category=category_button">
                <i>fa-solid fa-dollar-sign</i>
            </a>
        </div>
    </form>

```

Attribute Injection

The screenshot shows a browser developer tools interface with the Network tab selected. A single request is listed:

- Name:** store.php?search=%27+aut...
- Headers:** Content-Type: application/x-www-form-urlencoded
- Payload:** search=%27+auto...
- Response:** The response body contains the following HTML code:

```
<a class="add_button" href="neutemis.php"><i class="fa fa-solid fa-circle-plus"></i></a>
<div class="store_head">
  <input type="text" value="Search..." placeholder="Search..." name="search" value="%">
  <form class="search_bar" action="store.php" method="get">
    <div class="search_box">
      <input type="text" value="Search..." name="search" placeholder="Search..." value="%">
      <button type="submit"><i class="fa fa-solid fa-magnifying-glass"></i></button>
    </div>
  </form>
</div>
```
- Initiator:** (none)
- Timing:** 200 ms
- Cookies:** (none)

Below the Network tab, the status bar indicates 17 requests, 6.8 kB transferred, and 39 characters selected.

Image Error Handler

The screenshot shows the Network tab in the Chrome DevTools. A POST request to 'store.php?head=%22%3E...' is selected. The Response section displays the following payload:

```
<i class="fa-solid fa-circle-plus"></i>
<div class="store_head">...</div>
<div class="search_bar" action="store.php" method="get">
  <input type="text" name="search" placeholder="Search" value="&quot;&quot;&lt;img src=x onerror=alert(1)&gt;">
  ...
</div>
```

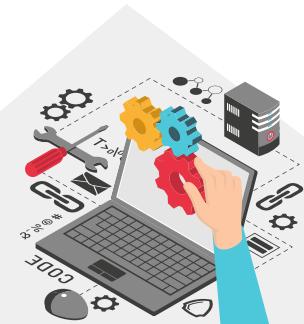
JavaScript URL



05.

SECURITY IMPLEMENTATION

(11. DIGITAL CERTIFICATE DEPLOYMENT)





Digital Certificate Deployment

The screenshot shows the Microsoft Edge browser interface with the following details:

- Address Bar:** https://localiKtech.com
- Page Title:** Privacy and security
- Left Sidebar (Privacy):**
 - Controls
 - Third-party cookies
- Left Sidebar (Security):**
 - Overview (selected)
 - Main origin
- Content Area:**

Security overview

This page is secure (valid HTTPS).

Certificate - valid and trusted
The connection to this site is using a valid, trusted server certificate issued by localiKtech.com.
[View certificate](#)

Connection - secure connection settings
The connection to this site is encrypted and authenticated using TLS 1.3, X25519, and AES_256_GCM.

Resources - all served securely
All resources on this page are served securely.
- Bottom Navigation:** Console, What's new, AI assistance, Issues, Developer resources, Filter by URL and error, Load through website
- Status Bar:** Status, URL, Initiator, Total Bytes, Duration, Error

Certificate Viewer: localkahtech.com

General Details

Issued To

Common Name (CN)	localkahtech.com
Organization (O)	Internet Widgits Pty Ltd
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	localkahtech.com
Organization (O)	Internet Widgits Pty Ltd
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Sunday, November 9, 2025 at 10:51 PM
Expires On	Monday, November 9, 2026 at 1:06:51 PM

SHA-256 Fingerprint

Certificate	0ace45cc86a3bed3ddae6f7a8bde586032c186aa513c5402b23529034 a520a
Public Key	5095e47730a68ced760dc196bb0fe1bd611dc3483e42623e39e17d120 bbc035

The screenshot shows the Network tab in the Chrome DevTools Performance panel. The timeline displays a sequence of requests, each with a color-coded initiator and a detailed breakdown of its duration across various stages. The legend at the top indicates the initiator type for each request: Preserve log (grey), Disable cache (blue), No throttling (green), and Load through website (orange). The bottom of the screenshot shows the Network conditions and Developer resources tabs selected.



Provides authenticated and encrypted communication, significantly improving overall data confidentiality and integrity.



05.

SECURITY IMPLEMENTATION

(12. CRYPTOGRAPHY IMPLEMENTATION)





Data-in-Transit (HTTPS/ TLS1.3)

Security overview

This page is secure (valid HTTPS).

 Certificate - valid and trusted
The connection to this site is using a valid, trusted server certificate issued by localkahtech.com.
[View certificate](#)

 Connection - secure connection settings
The connection to this site is encrypted and authenticated using TLS 1.3, X25519, and AES_256_GCM.

 Resources - all served securely
All resources on this page are served securely.

TLS 1.3 setup ensures all traffic fully encrypted and protected in transit, keeping sensitive data secure without needing extra application-level encryption.



Data-at-Rest (AES-256-GCM)

Before & After:

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	None
2	user_name	varchar(255)	utf8mb4_general_ci		No	None
3	email	varchar(255)	utf8mb4_general_ci		No	None
4	secondary_email	varchar(255)	utf8mb4_general_ci	Yes	NULL	
5	phone	varchar(15)	utf8mb4_general_ci		No	None
6	user_address	varchar(255)	utf8mb4_general_ci		No	None
7	pwd	varchar(255)	utf8mb4_general_ci		No	None
8	user_image	varchar(255)	utf8mb4_general_ci	Yes	no_profile_pic.png	
9	wrong_pwd_count	int(11)			Yes	0
10	lock_until	datetime			Yes	NULL

Key:

```
<?php
echo bin2hex(string: random_bytes(length: 32));
?>
```

localhost/INT6005CEM_security_group_assignment_G23/Online-Course-Management-System

```
60d4d9476c34544becd52ff72cdfaec819202e5d715b97648f723c69cdeb32ec
```

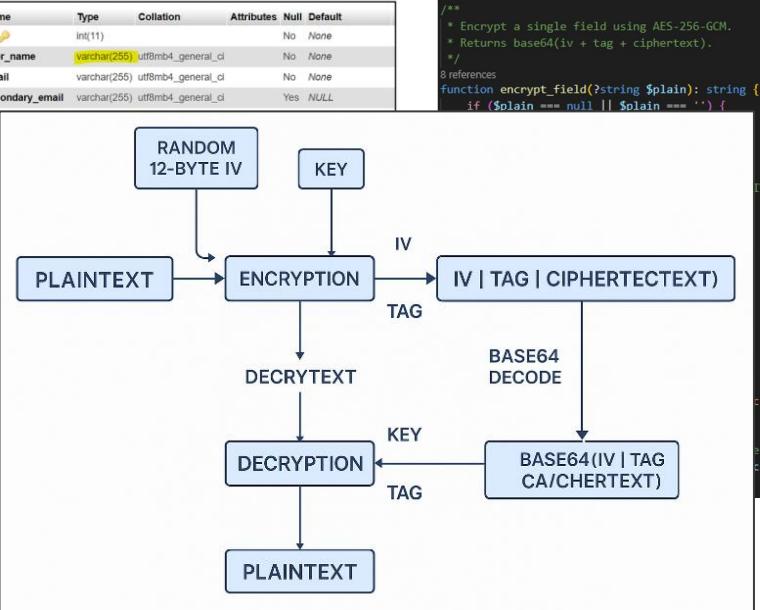
```
// 32-byte encryption key
// Generate in keygen.php
const ENC_KEY_HEX = '60d4d9476c34544becd52ff72cdfaec8192';

// Convert hex to binary key
const ENC_METHOD = 'aes-256-gcm';
```

2 references

```
function get_enc_key(): string {
    return hex2bin(string: ENC_KEY_HEX);
}
```

Encryption:



```
/*
 * Encrypt a single field using AES-256-GCM.
 * Returns base64(iv + tag + ciphertext).
 */
8 references
27 references
function encrypt_field(?string $plain): string {
    if ($plain === null || $plain === '') {
        return '';
    }

    $blob = base64_decode(string: $encoded, strict: true);
    if ($blob === false || strlen(string: $blob) < 12 + 16) {
        return '';
    }

    $iv = substr(string: $blob, offset: 0, length: 12);
    $tag = substr(string: $blob, offset: 12, length: 16);
    $cipher = substr(string: $blob, offset: 28);

    plain = openssl_decrypt(
        data: $cipher,
        cipher algo: ENC_METHOD,
        passphrase: get_enc_key(),
        options: OPENSSL_RAW_DATA,
        iv: $iv,
        tag: $tag
    );

    return $plain === false ? '' : $plain;
}
```



Data-at-Rest (AES-256-GCM)

Code-Based Verification Against Database Value

Test: AES-256-GCM decryption

User #28 – Ensin

Compares the encrypted values in the database with the decrypted output and the expected real values (username, email, phone, address).

Field	Database value (encrypted)	After decrypt_field()	Expected real value	Result
username	Nwksly+F+HZG1TUS14a1A+p0suRf9lRDcCwztNUJ6n1CnAs	Ensin	Ensin	MATCH
email	K5E\oCjzEmBcd0yZb6R21IVD0QZ4N51WLHRjXz6NTH8UH2bfU6rExN29FBtbH2Dw=	ensin2004@gmail.com	ensin2004@gmail.com	MATCH
secondary_email		-		Not tested
phone	Goh5s1e/YLvvLq117sBbKqjVGwdLcnusabG9stPDMd8QaDzR1E=	0193818893	0193818893	MATCH
user_address	Ksy1nTg6peXQphszZ+z890sD0PnY8rY8wEvh0wAbNR5Fc6fRM1vp1vM9w==	ensin is kawaii	ensin is kawaii	MATCH

● MATCH – decrypted value is exactly the same as the expected real value.
● NOT MATCH – decrypted value is different from the expected value.
● Not tested – no expected value provided (e.g. secondary email not set).

UI-Level Check

My Account

Name :

Email :

Phone Number :

Address :

New Password :

Confirm Password :

I confirm that I have read and agree to KAH TECH
 [User Agreement](#) and [Privacy & User Education Policy](#).

[Sign up](#)

Already have an account? [Log In](#).

[EDIT](#) [LOG OUT](#)



AES-256-GCM setup ensures sensitive database fields remain unreadable if leaked while still decrypting safely and accurately when the system needs them.



05.

SECURITY IMPLEMENTATION **(13. BROWSER COOKIE)**





Browser Cookie Security

```
session_set_cookie_params([
    'lifetime' => 0,           // expires when browser closes
    'path' => '/',
    'secure' => true,          // only over HTTPS
    'httponly' => true,        // JS cannot access it
    'samesite' => 'Strict' // strong CSRF protection
]);
```

Browser cookies are configured with strict security settings to protect from theft, sniffing and CSRF attacks

The screenshot shows a browser window with developer tools open, specifically the Application tab. The page content displays a dark-themed website with sections for 'About Us' and 'Customize Your Own'. In the developer tools, the Application tab lists various storage components like Manifest, Service workers, and Storage. Under Storage, the Cookies section is expanded, showing a table of cookies. One cookie, 'PHPSESSID', is highlighted with a red border. The table columns include Name, Value, Domain, Path, Expires, Size, HttpOnly, Secure, SameSite, Partition, Cross-Site, and Priority. The 'Value' column for the highlighted cookie shows the value '2glikoja311324h5oi8qehai54e'. At the bottom of the developer tools, the 'Cookie Value' field also displays '2glikoja311324h5oi8qehai54e'.

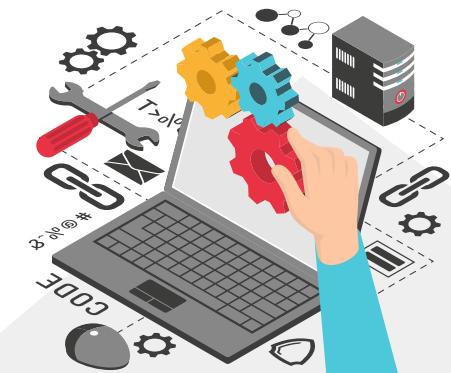
Name	Value	Dom...	Path	Expires...	Size	Http...	Secure	Same...	Partit...	Cross...	Priority
PHPSESSID	2glikoja311324h5oi8qehai54e	local...	/	Sessi...	35	✓	✓	Strict			Medium





06.

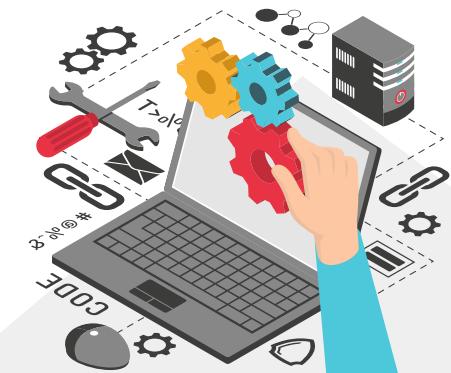
DISCUSSION & SUMMARY





06.

DISCUSSION & SUMMARY (DISCUSSION)





OWASP TOP 10 COVERAGE

ACHIEVED	PARTIALLY ACHIEVED	NOT APPLICABLE
<p>A01 – Broken Access Control</p> <p>A02 – Cryptographic Failures</p> <p>A03 – Injection (SQLi)</p> <p>A05 – Security Misconfiguration</p> <p>A07 – Identification & Authentication Failures</p> <p>A09 – Security Logging & Monitoring Failures</p>	<p>A04 – Insecure Design</p> <p>A06 – Vulnerable & Outdated Components</p> <p>A08 – Software & Data Integrity Failures</p>	<p>A10 – Server-Side Request Forgery (SSRF)</p>



REMAINING VULNERABILITIES AFTER CONTROL

A01

Broken Access Control

- New pages/APIs may skip role checks
- Risk of exposing sensitive IDs in URLs
- **Fix:** Follow consistent access control rules

A03

Injection

- Legacy code may use unsafe SQL
- **Fix:** Enforce prepared statements in code reviews



A02

Cryptographic Failures

- AES/TLS depends on proper key storage
- Require encryption and strict access to backups
- **Fix:** Improve key handling & backup encryption

A04

Insecure Design

- Controls exist but no formal threat modelling
- **Fix:** Use simple design/threat checklist



REMAINING VULNERABILITIES AFTER CONTROL

A05

Security Misconfiguration

- Configurations may drift during deployment
- New servers may miss error pages or HSTS
- **Fix:** Use deployment checklists to prevent drift

A07

Identification & Authentication Failures

- Strong passwords + rate limiting exist
- Still vulnerable to credential stuffing and no MFA for admins
- **Fix:** Enforce MFA for admins; offer MFA for users



A06

Vulnerable & Outdated Components

- Libraries may become outdated silently
- No automated dependency monitoring
- **Fix:** Add automated scanning and patch cycles

A08

Software & Data Integrity Failures

- No code-signing or pipeline integrity checks
- **Fix:** Harden DevOps pipeline and use signed releases



REMAINING VULNERABILITIES AFTER CONTROL

A09

Security Logging & Monitoring Failures

- Logs not analysed centrally
- Suspicious activity may be detected late
- **Fix:** Centralize logs / SIEM

A10

SSRF

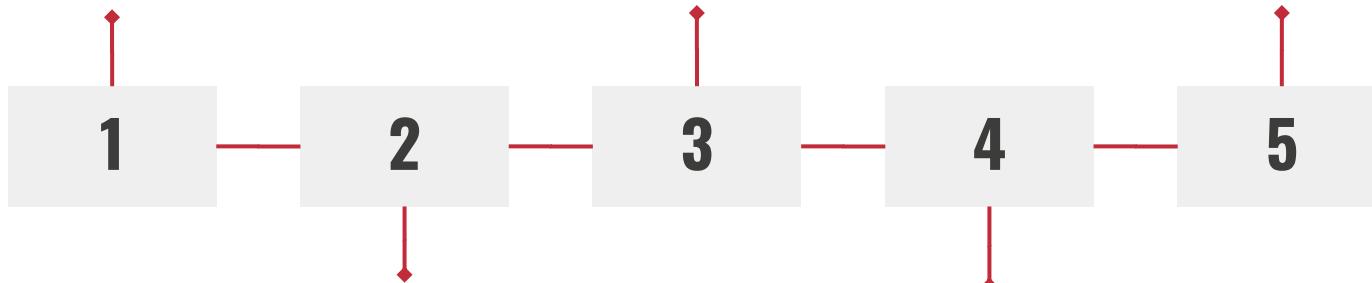
- Not currently applicable
- Future features (webhooks, URL fetchers) could introduce SSRF
- **Fix:** Apply SSRF-safe design for URL-fetching features



FUTURE ENHANCEMENT

Full MFA for Admins

- Enable MFA for all admin login
- Use TOTP + controlled recovery
- Reduces password compromise



Automated Dependency Scanning

- Track all libraries/component
- Use Composer/NPM/Git scans
- Catch CVEs before deployment

Secure SDLC & Checklists

- Use security checklists in reviews
- Quick threat-modelling for major changes
- Prevents new vulnerabilities



CSP & Browser Hardening

- Add CSP to block untrusted scripts
- Set security headers (nosniff, DENY, referrer policy)
- Mitigates XSS + clickjacking

Centralized Logging & Alerts

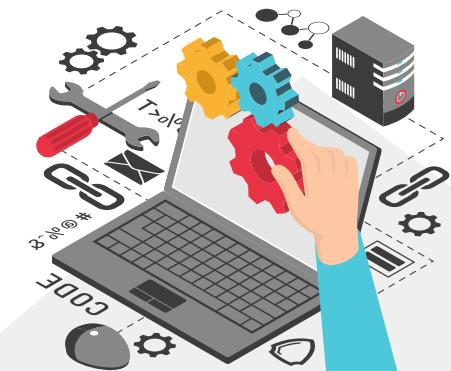
- Send app + server logs to SIEM/ELK
- Dashboards + alerts for anomalies
- Faster threat detection





06.

DISCUSSION & SUMMARY (SUMMARY)



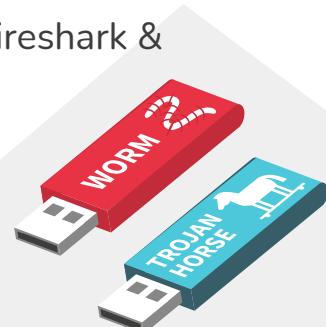
SUMMARY

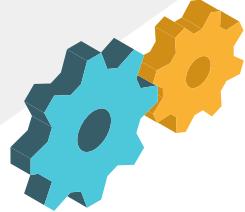
CONCLUSION

- System now addresses most OWASP Top 10 risks
- Major OWASP risks addressed (SQLi, XSS, weak sessions, no TLS, plaintext passwords)
- Implemented Argon2id, TLS 1.3, CSRF, secure cookies, logging & safer queries
- Overall system now far more secure and closer to production-ready

LEARNING OUTCOME ACHIEVED

- Hands-on experience with SQLi, XSS, session abuse & crypto issues
- Implemented real defenses: Argon2id, AES-GCM, TLS, CSRF, secure sessions
- Improved risk prioritization (Likelihood × Severity)
- Gained skills with Burp, Wireshark & dev tools





THANK YOU

