UKM ONLINE Q&A PLATFORM

LI AONAN, PROF.DR. ZARINA SHUKUR

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY UNIVERSITI KEBANGSAAN MALAYSIA
43600 BANGI, SELANGOR

ABSTRAK

Platform Soal Jawab Dalam Talian UKM bertujuan untuk menghapuskan jurang komunikasi yang terpisah dan kekurangan maklumat yang boleh dipercayai bagi pelajar, terutamanya pelajar tahun pertama. Pendaftaran kursus, pengurusan asrama, dan prosedur pentadbiran sering menjadi cabaran kepada pelajar yang terpaksa menggunakan saluran tidak rasmi seperti media sosial, yang tidak disesuaikan untuk memenuhi keperluan mereka.Projek ini akan membangunkan sebuah platform web berpusat menggunakan HTML, Python, dan MySQL. Platform ini akan menyediakan ruang komunikasi yang teratur, membolehkan pelajar dan pensyarah bertanya dan menjawab soalan, serta menyediakan perpustakaan Soalan Lazim (FAQ) yang boleh diakses oleh pelajar dan pensyarah. Dengan menyatukan maklumat, platform ini akan menyederhanakan dan mempercepatkan komunikasi antara pelajar dan pensyarah, menjadikan akses kepada maklumat yang tepat lebih pantas dan mudah, serta menggalakkan kerjasama antara pelajar dan pensyarah.Platform ini akan menampilkan fungsi-fungsi utama seperti memposting, menjawab soalan, dan perpustakaan FAQ yang disusun mengikut topik, serta pilihan perundingan dengan pensyarah. Struktur ini memudahkan pelajar mencari jawapan dengan pantas, berbincang dengan pensyarah, dan berinteraksi dengan rakan sebaya. Manfaat utama yang diperolehi termasuk sokongan akademik yang lebih baik, akses kepada maklumat yang boleh dipercayai, hubungan yang lebih kukuh dalam komuniti, dan tahap kepuasan pelajar yang lebih tinggi.

ABSTRACT

The UKM Online Q&A Platform eliminates the gap between fragmented communication and complete lack of reliable information for students, especially first year learners. Course registration, hostel arrangements and administrative procedures create a problem for students who find themselves using inferior channels like social

media that isn't tailored to suit their unique needs. The project is to develop a centralized web platform using PHP-Python hybrid system, HTML, CSS, JavaScript, MySQL with advanced indexing. This will create an organized place for communication where it will allow students and faculty to ask questions, answers and FAQ can be provided by the students and faculty. By consolidating information, the platform will simplify and speed up communication between students and faculty alike, at the same time make access to accurate information quick and convenient, and encourage collaboration between students and faculty. The platform will feature such key functions as posting, answering, and FAQ libraries arranged by topic as well as consultation options to faculty. This structure facilitates quick search by students for answers, discussions with lecturers and interactions with peers, the main benefits of which include better academic support, access to reliable information, stronger connections with the community and higher levels of student satisfaction.

1.0 INTRODUCTION

In today's technologically advanced and ever-changing world, enhancing communication and knowledge management is essential for academic institutions. Like most higher education institutions in Malaysia, Universiti Kebangsaan Malaysia (UKM) lacked a one-stop hub to connect students and lecturers. For this reason, these gaps cause disjointed information sharing, misconceptions, and loss of potential learning synergies. The main rationale leading to the creation of a Q&A website that is only accessible by the members of the UKM originates from the experiences of the university's members in finding and acquiring up-to-date and correct information. This is unlike the platforms like Quora or Zhihu, where people feel the need of a secure, member-only confined environment for the discussion. To achieve privacy and relevance, The platform will be secured using an email-based authentication system and users must register using a UKM email address (e.g. @siswa.ukm.edu.my or @ukm.edu.my). The relevance of this approach can be explained with help of the fact that more and more people use online resources in order to get the support in educational process as well as to find their groups of like-minded people. A dedicated Q&A website for UKM would: Enhance Learning Opportunities: It becomes convenient for students to ask questions concerning any academic issue, probably discuss their programs, and

work on assignments. Improve Communication Efficiency: By doing so, lecturers and officers are able to handle similar queries and questions within a public fashion thus eliminating constant questions but also guarantee an accurate answer. Foster Community Engagement: In this way, fostered through the platform, the sense of belonging to the university and mutual support of peers is actively developed. From a survey of literature on digital learning environment, institution specific platforms play a very crucial role in enhancing student learning and performance (Smith & Johnson, 2020). In addition, there has been evidence that shows that integration of such platforms will reduce paperwork and enhance satisfaction amongst the university stakeholders (Lee et al., 2019).

It is important to set up effective communication and knowledge sharing practices as two important characteristics of a productive academical community. At the moment, there is no centralized and secure virtual ground in the Universiti Kebangsaan Malaysia (UKM) available for healthy communication between student-lecturer-administrator. Such lack raises several questions that shape the academic climate and organizational effectiveness of the university.(Gasmi 2022)

Firstly, there is what might be referred to as the decentralization of communication, where the multiple segments have made it difficult to establish a common ground. Some of these communication methods include and email, use of unofficial social media group and word of mouth among the students and faculty. Such decentralisation results in misunderstanding, slow dissemination of information, and refusal propagation of true information. For example, important announcements or updates shared through unofficial channels can be missed or misinterpreted, leading to confusion and inefficiency(Liu, Wang et al. 2020)

Secondly, the students lacks some academic assistance once in the classroom. This is because without a centralised Q&A platform it is difficult for students to find someone to ask help on any issues they encounter in their academics. This eliminates grouping of students and peer collaboration which are important in enhancing understanding and improvement of performance. Research has shown that online academic forums enhance student engagement and learning outcomes by providing a

space for ongoing academic discussion (Garrison and Cleveland-Innes 2005, Hrastinski 2009)

Third, the administrative systems in the management of admission centre are ineffectual in regard to its students' as well as administrative functions. Students' administration issues concerning registration, visa extensions, and other campus issues that the officials deem minor, often meet teenage students with unclear procedures without proper direction. The lack of a streamlined channel to address these issues can lead to frustration and potential delays in critical processes, negatively impacting the overall student experience (DeAndrea, Ellison et al. 2012)

However, engaging current public forums and social media sites present a concern in security and privacy. Such platforms have no adequate measures of safeguarding sensitive information regarding the university. This vulnerability poses a risk to personal data and institutional integrity as unauthorised access or data breaches can have serious consequences (Alwi and Fan 2010) Security and Privacy Issues are anonymous and open social media applications that portend the current social media trends lack mechanisms of preventing leakage of sensitive information that may be university related.

Proposed Solution

Create a secure, university-specific Q&A platform: Use UKM-specific email addresses (@siswa.ukm.edu.my and @ukm.edu.my) for user access verification, providing a secure internal communication space for UKM members.

Improve user accessibility and personalized profiles: Each user has their own homepage, which includes a record of questions and answers posted on the site, as well as a history of their activity.

Improve information discovery through advanced search and tagging: Support search and tagging features, allowing users to quickly find information using keywords, tags, or categories.

Ensure mobile accessibility and user convenience: Compatible with mobile, web, and desktop devices.

Real-time notifications: Receive instant notifications whenever a user has a new reply.

Project Objectives (SMART):

S (Specific): Create a UKM Q&A website, including a Q&A area where students, lecturers, and administrators can find information, ask questions, and receive answers.

M (Measurable): Implement a secure authentication system using UKM email verification; achieve 90% functionality for end users; achieve 85% search accuracy; achieve ≥95% real-time alert accuracy; and achieve at least 50 registered users within three months.

A (Achievable): Develop using common tools and technologies, including a MySQL database, HTML/CSS/JavaScript front-end, and PHP back-end.

R (Relevant): Address communication and administrative issues identified by UKM, enhance academic support, and ensure confidentiality of student information.

T (Timeframe): Complete within one year, divided into four phases: Planning and Requirements Gathering (3 months), Platform Architecture Development (4 months), Core Functionality Development (3 months), and Testing and Deployment (2 months).

1.1 SYSTEM USE CASE DIAGRAM

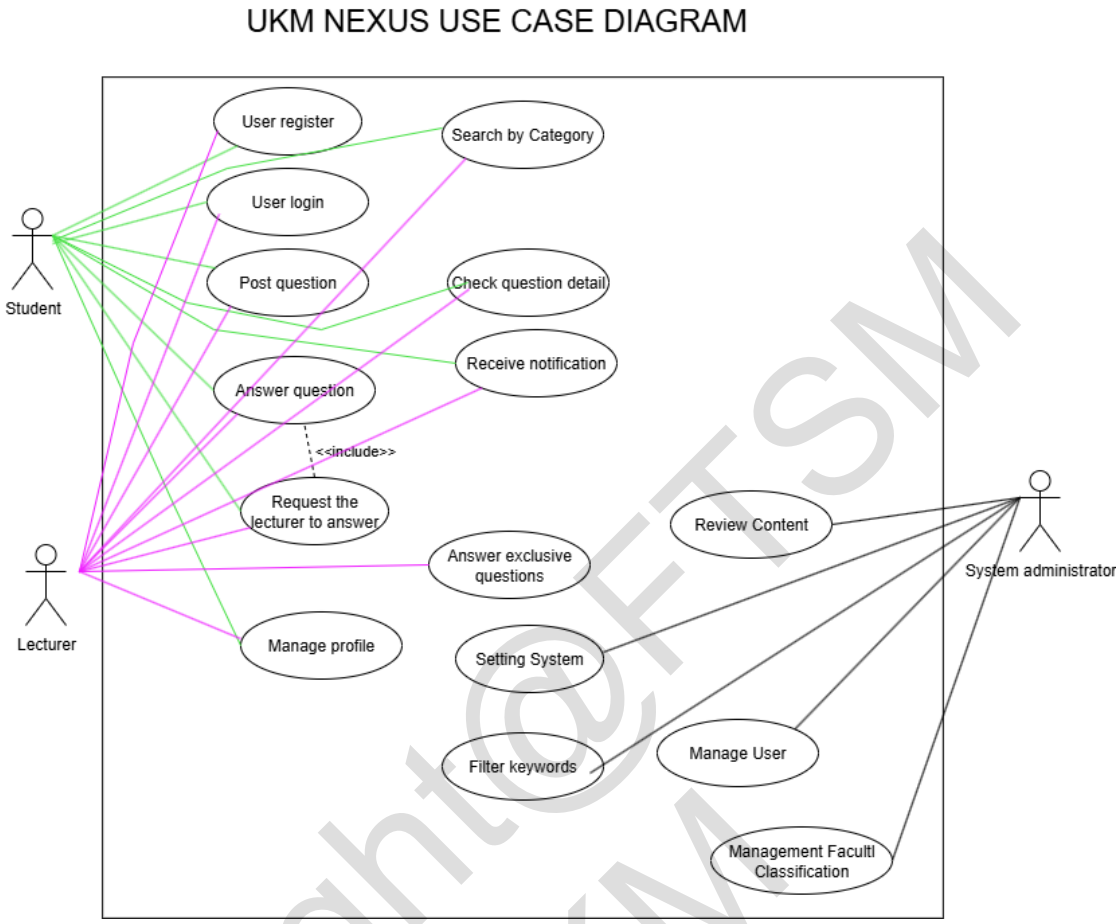


Figure 1.1 Use Case diagram

1.2 SYSTEM DEVELOP FLOW CHART

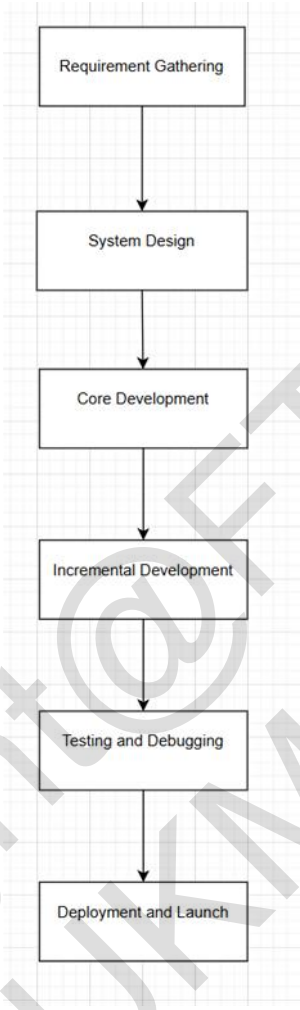


Figure 1.2 System develop flowchart

1.3 WORK BREAKDOWN STRUCTURE (WBS)

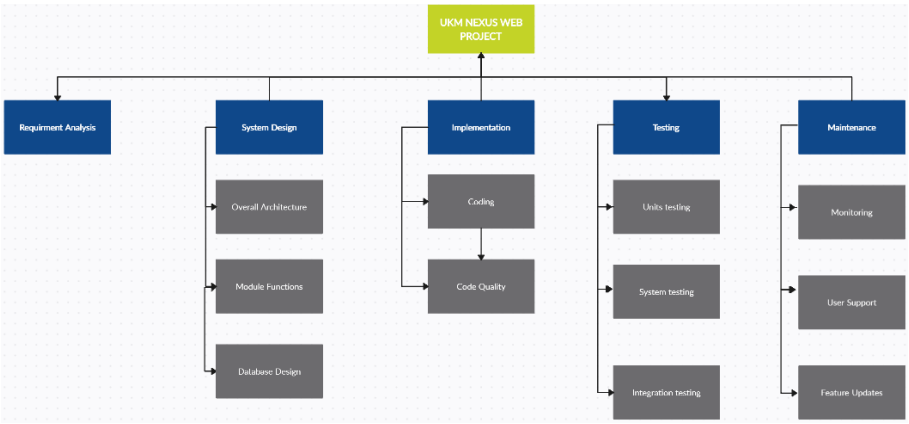


Figure 1.3 work breakdown structure

2.1 LITERATURE REVIEW

Table 2.1 Compare and contrast table

Platform	Problem Solution	Advantage	Disadvantage
UKM Q&A Platform	University-specific academic Q&A	Privacy, only UKM community, targeted academic support	Limited to UKM, smaller user base compared to global platforms
Quora	Question-and-answer with expert contributors	Reliable information, expert insights, quality answers	Varies by contributor, less focus on niche topics
Tumblr	Microblogging, creative content sharing	Emphasis on creativity, strong subcultures	Limited structured discussion, not suitable for formal Q&A
Hacker News	Tech-focused news and discussions	High-quality tech discussions, niche community	Overly technical discussions, limited general audience appeal
Stack Exchange	Structured Q&A for specialized topics	Accurate, curated responses, reputation system	Strict guidelines, intimidating for new users
Reddit	Community-driven with subreddits	Diverse content, large user base, high engagement	Inconsistent moderation, varying content quality

Each platform has its own unique characteristics and target customer groups. The UKM Q&A platform is designed specifically for universities, ensuring academic assistance and collaboration between students, lecturers, and administrators. Compared to global platforms, UKM Q&A provides a private, focused, and academically oriented environment.

3.1 METHODOLOGY

Development was conducted using an incremental development model. This multi-tiered approach enabled the platform to be built and deployed in phases, with features developed, tested, and released incrementally.

Development Phase

Requirements Analysis: Clearly understand user requirements for the UKM Q&A website, including user authentication with UKM email, the ability to post and answer questions, search functionality, real-time notifications, and user activity tracking profiles.

System Design: Develop the overall system architecture, defining the functionality of each module and how they interact, including database design, data relationship definition, and authentication/posting/search/notification workflow design.

Implementation: Translate the design into a functional platform and write application code using current programming languages and frameworks, ensuring clean, maintainable, and extensible code.

Testing: Includes key phases such as unit testing, integration testing, system testing, and user acceptance testing (UAT).

Maintenance: Continuously monitor the system, resolve issues, add new features based on user feedback, apply security updates, and fix bugs.

3.2 NEEDS ANALYSIS

Centralized and reliable information source: 70% of students report being frustrated by fragmented information and primarily use social media as a source of information. The platform provides a single portal for students to browse verified and organized campus content.

Academic and administrative support:

Lecturer advice: Students have a strong demand for academic guidance and want to receive advice directly from their lecturers.

Administrative inquiries: First-year students, in particular, face challenges such as finding a classroom and renewing their student visas.

Web-based accessibility: A survey shows that 95.7% of students prefer the website to mobile apps.

Daily life functionality: 91.3% of students would like recommendations for affordable food options on or near campus.

Ease of information search and navigation: Students need intuitive search and tagging functionality to quickly and efficiently find information on specific topics, such as faculty locations or administrative services.

3.2.1 User Data Analysis

User feedback questionnaires were used to understand key user needs and concerns:

Need for a dedicated Q&A website: 89.3% believe it's necessary, 7.1% are unsure, and 3.6% don't think it's necessary.

Information acquisition channels: Social media (57.1%), classmates and friends (17.9%), school bulletin boards (14.3%).

Information acquisition challenges: 67.9% lack of consistent data, 21.4% insufficient materials, and 10.7% privacy and security concerns.

Desired features: 50% seek instructor advice, 21.4% seek quick answers to academic questions, and 21.4% want to share learning resources.

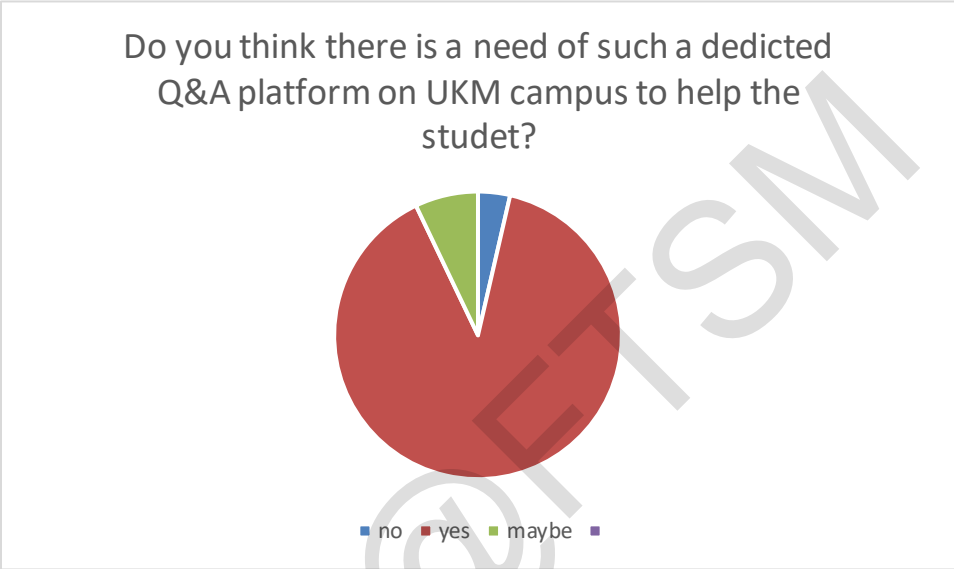


Figure 3.1 UKM Nexus google form

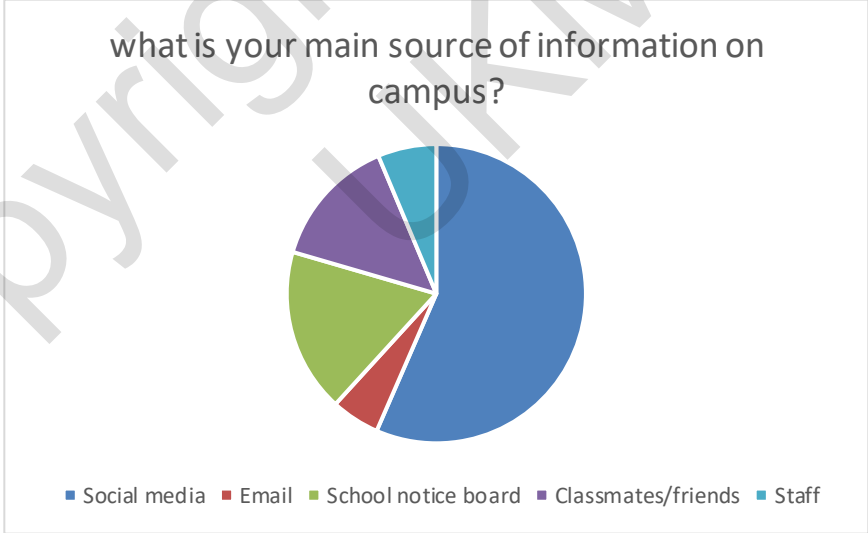


Figure 3.2 UKM nexus google form

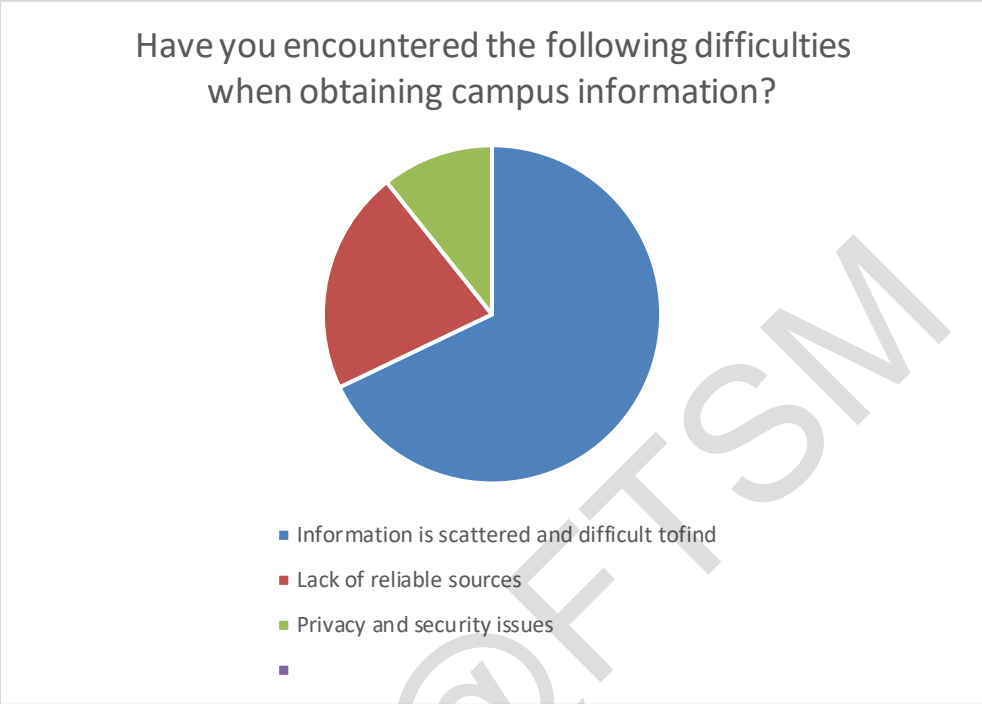


Figure 3.3 UKM nexus google form

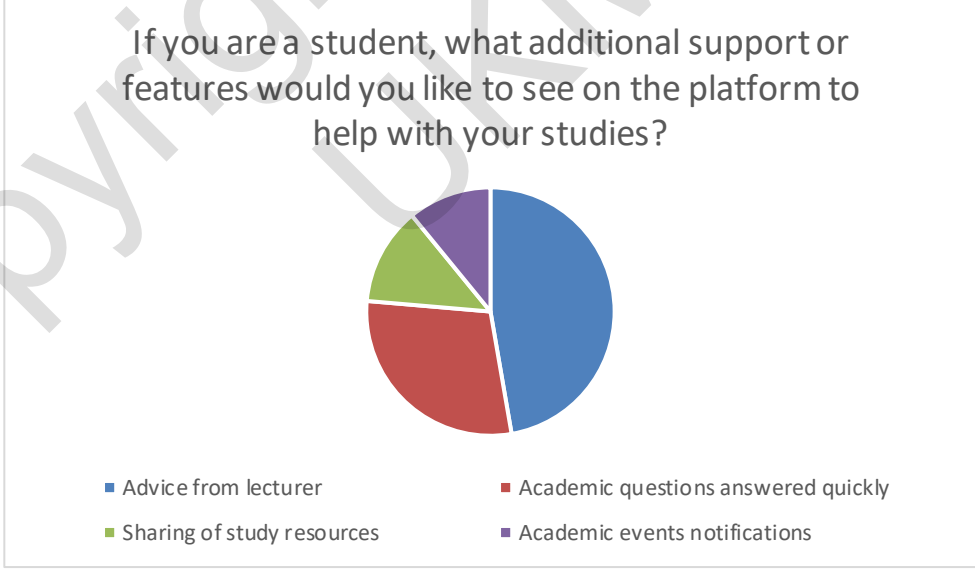


Figure 3.4 UKM nexus google form

3.3 CONCEPTUAL MODEL DESIGN

3.3.1 System Architecture Design

The Model-View-Controller (MVC) architectural pattern is used to ensure separation of concerns and maintainability.

User Management Module:

Purpose: User management, authentication, and role determination based on UKM email authentication

Relationship: Interacts with the Database Module for user login

Q&A Module:

Purpose: Enables users to ask questions, answer questions, and read others' answers

Relationship: Connects the Search Module and Notification Module

Search Module:

Purpose: Implements advanced search and tagging capabilities to locate questions, answers, or categories

Relationship: Retrieves and filters data from the Database Module, using tagging rules provided by the Administration Module

Notification Module:

Purpose: Sends instant notifications about new answers, announcements, and system updates

Relationship: Interacts with the User Management and Q&A Module

Admin Module:

Purpose: Content moderation, platform management, and announcement publishing

Relationship: Interacts with all other modules for administrative control

Database Module:

Purpose: Maintains user profiles, questions, answers, tags, and notification logs

Relationship: Provides backend support for all other modules

Network Topology Design

Implements a client-server model combined with a hub-and-spoke topology. Includes phpMyAdmin for improved database management and control.

4.1 WEB DEVELOPMENT

4.1.1 development environment

The UKM NEXUS platform was developed using a PHP-based technology stack with MySQL for database management. The development environment was set up as follows:

1. XAMPP: Used as the local development server environment
2. Visual Studio Code: Employed as the code editor
3. MySQL : Utilized for database design and management
4. GitHub: Implemented for version control

4.1.2 User Authentication Module

The user authentication module implements secure registration and login functionality with email verification using the UKM email domain. This was a critical security component to ensure only legitimate UKM community members can access the platform.

1. Email Verification System

```python
import smtplib
import sys
import random
import string
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.header import Header

def generate_verification_code(length=6):

    characters = string.digits #只限数字
    return ''.join(random.choice(characters) for _ in range(length))

def send_email(recipient_email, smtp_server, smtp_port, sender_email, sender_password):
    """收件人邮箱地址,服务器,端口,发送人的邮箱,发送的密码"""
    verification_code = generate_verification_code()

    # 邮件内容
    html_content = f"""
    <html>
    <head>
        <style>
            body {{ font-family: 'Arial', sans-serif; color: #333; }}
            .container {{ max-width: 600px; margin: 0 auto; padding: 20px; }}
            .header {{ background-color: #8E44AD; color: white; padding: 20px; text-align: center; }}
            .content {{ padding: 20px; background-color: #f9f9f9; }}
            .code {{ font-size: 32px; font-weight: bold; text-align: center; color: #8E44AD;
                    margin: 20px 0; letter-spacing: 5px; }}
            .footer {{ text-align: center; margin-top: 20px; font-size: 12px; color: #888; }}
        </style>
    </head>
    <body>
        <div class="container">
            <div class="header">
                <h1>UKM NEXUS Email Verification</h1>
            </div>
            <div class="content">
                <p>Dear UKM Student,</p>
                <p>Thank you for registering with UKM NEXUS. To complete your registration, please use the verification code below:</p>
                <div class="code">{verification_code}</div>
                <p>This code will expire in 10 minutes.</p>
                <p>If you did not request this code, please ignore this email.</p>
            </div>
            <div class="footer">
                <p>This is an automated message, please do not reply to this email.</p>
                <p>&copy; {2025} UKM NEXUS. All rights reserved.</p>
            </div>
        </div>
```

Figure 4.1 Python Language to Connect PHP

```python
    # 创建邮件对象
    message = MIMEMultipart('alternative')
    message['From'] = sender_email
    message['To'] = recipient_email
    message['Subject'] = 'UKM NEXUS - Email Verification Code'

    # 添加HTML内容
    message.attach(MIMEText(html_content, 'html', 'utf-8'))

    try:
        smtp_obj = smtplib.SMTP_SSL(smtp_server, smtp_port)  # 使用SSL连接
        smtp_obj.login(sender_email, sender_password)
        smtp_obj.sendmail(sender_email, [recipient_email], message.as_string())
        smtp_obj.quit()
        return verification_code, "success"
    except Exception as e:
        return None, str(e)

if __name__ == "__main__":

    # 从命令行参数获取数据
    if len(sys.argv) != 6:
        print("Error: Incorrect number of arguments")
        print("Usage: python send_verification_email.py <recipient_email> <smtp_server> <smtp_port> <sender_email> <sender_password>")
        sys.exit(1)

    recipient_email = sys.argv[1]
    smtp_server = sys.argv[2]
    smtp_port = int(sys.argv[3])
    sender_email = sys.argv[4]
    sender_password = sys.argv[5]

    verification_code, status = send_email(recipient_email, smtp_server, smtp_port, sender_email, sender_password)

    # 输出的结果方便PHP读取
    if verification_code:
        print(f"{verification_code}|success")
    else:
        print(f"error|{status}")
```

Figure 4.2 Python Language to Connect PHP

4.1.3 User Registration Implementation

```php
<?php
session_start();
require_once 'includes/config.php';

if(isset($_SESSION['user_id'])) {
    header(header: "Location: forum.php");
    exit();
}
//smtp 邮件设置
$smtp_server = "smtp.gmail.com";
$smtp_port = 465;  // SSL端口
$sender_email = "aslm2lik123@gmail.com";
$sender_password = "lcfypbwumaufqnlj";  //

$error = '';
$success = '';

// 处理验证码发送请求
if (isset($_POST['send_verification_code'])) {
    $email = trim(string: $_POST['email']);

    // 验证邮箱格式
    if (!preg_match(pattern: '/^[a-zA-Z0-9]+@siswa\.ukm\.edu\.my$/', subject: $email)) {
        echo json_encode(value: ['status' => 'error', 'message' => 'Please use your UKM student email (@siswa.ukm.edu.my).']);
        exit;
    }

    // 检查邮箱是否已注册
    $stmt = $pdo->prepare(query: "SELECT id FROM users WHERE email = ?");
    $stmt->execute(params: [$email]);
    if ($stmt->rowCount() > 0) {
        echo json_encode(value: ['status' => 'error', 'message' => 'This email is already registered.']);
        exit;
    }

    // 调用Python脚本发送验证码
    $command = escapeshellcmd(command: "python3 send_verification_email.py " .
               escapeshellarg(arg: $email) . " " .
               escapeshellarg(arg: $smtp_server) . " " .
               escapeshellarg(arg: $smtp_port) . " " .
               escapeshellarg(arg: $sender_email) . " " .
               escapeshellarg(arg: $sender_password));

    $output = shell_exec(command: $command);
    $result = explode(separator: "|", string: $output);
```

Figure 4.3 Set Smtp Connect API to send Email

```php
    if (isset($result[0]) && $result[0] != "error") {
        $verification_code = trim(string: $result[0]);

        // 检查数据库中是否已有验证码表
        try {
            $pdo->query(query: "SELECT 1 FROM verification_codes LIMIT 1");
        } catch (PDOException $e) {
            //没有就创建
            $pdo->exec(statement: "CREATE TABLE verification_codes (
                id INT AUTO_INCREMENT PRIMARY KEY,
                email VARCHAR(100) NOT NULL UNIQUE,
                code VARCHAR(10) NOT NULL,
                expiry_time DATETIME NOT NULL,
                created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
            )");
        }

        // 验证码时间
        $expiry_time = date(format: 'Y-m-d H:i:s', timestamp: strtotime(datetime: '+10 minutes'));

        // 检查是否已有此邮箱的验证码记录
        $stmt = $pdo->prepare(query: "SELECT id FROM verification_codes WHERE email = ?");
        $stmt->execute(params: [$email]);

        if ($stmt->rowCount() > 0) {
            // 更新现有记录
            $stmt = $pdo->prepare(query: "UPDATE verification_codes SET code = ?, expiry_time = ? WHERE email = ?");
            $stmt->execute(params: [$verification_code, $expiry_time, $email]);
        } else {
            // 插入新记录
            $stmt = $pdo->prepare(query: "INSERT INTO verification_codes (email, code, expiry_time) VALUES (?, ?, ?)");
            $stmt->execute(params: [$email, $verification_code, $expiry_time]);
        }

        echo json_encode(value: ['status' => 'success', 'message' => 'Verification code sent to your email.']);
    } else {
        $error_message = isset($result[1]) ? trim(string: $result[1]) : "Failed to send verification code.";
        echo json_encode(value: ['status' => 'error', 'message' => $error_message]);
    }
    exit;
}
```

Figure 4.4 Set Smtp Connect API to send Email

```php
// 验证校验
if (isset($_POST['verify_code'])) {
    $email = trim(string: $_POST['email']);
    $code = trim(string: $_POST['code']);

    // 检查是否正确
    $stmt = $pdo->prepare(query: "SELECT code, expiry_time FROM verification_codes WHERE email = ?");
    $stmt->execute(params: [$email]);

    if ($stmt->rowCount() == 0) {
        echo json_encode(value: ['status' => 'error', 'message' => 'Please request a verification code first.']);
        exit;
    }

    $row = $stmt->fetch(mode: PDO::FETCH_ASSOC);
    $db_code = $row['code'];
    $expiry_time = $row['expiry_time'];

    // 时间是否过期
    if (strtotime(datetime: $expiry_time) < time()) {
        echo json_encode(value: ['status' => 'error', 'message' => 'Verification code has expired. Please request a new one.']);
        exit;
    }

    // 检查验证码是否正确
    if ($code != $db_code) {
        echo json_encode(value: ['status' => 'error', 'message' => 'Invalid verification code.']);
    } else {
        echo json_encode(value: ['status' => 'success', 'message' => 'Email verified successfully.']);
    }
    exit;
}
```

Figure 4.5 Set Smtp Connect API to send Email

```php
// 处理注册请求
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['username'])) {
    $username = trim(string: $_POST['username']);
    $email = trim(string: $_POST['email']);
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    $verification_code = isset($_POST['verification_code']) ? trim(string: $_POST['verification_code']) : '';

    // Validate user input
    if (empty($username) || empty($email) || empty($password) || empty($confirm_password) || empty($verification_code)) {
        $error = "All fields are required.";
    } elseif (strlen(string: $username) < 3) {
        $error = "Username must be at least 3 characters.";
    } elseif (!preg_match(pattern: '/^[a-zA-Z0-9]+@siswa\.ukm\.edu\.my$/', subject: $email)) {
        $error = "Please use your UKM student email (@siswa.ukm.edu.my).";
    } elseif (strlen(string: $password) < 6) {
        $error = "Password must be at least 6 characters.";
    } elseif ($password !== $confirm_password) {
        $error = "Passwords do not match.";
    } else {
        // 验证验证码
        $stmt = $pdo->prepare(query: "SELECT code, expiry_time FROM verification_codes WHERE email = ?");
        $stmt->execute(params: [$email]);

        if ($stmt->rowCount() == 0) {
            $error = "Please verify your email first.";
        } else {
            $row = $stmt->fetch(mode: PDO::FETCH_ASSOC);
            $db_code = $row['code'];
            $expiry_time = $row['expiry_time'];

            // 检查验证码是否过期
            if (strtotime(datetime: $expiry_time) < time()) {
                $error = "Verification code has expired. Please request a new one.";
            }
            // 检查验证码是否正确
            elseif ($verification_code != $db_code) {
                $error = "Invalid verification code.";
            } else {
                // 检查用户名或邮箱是否已被使用
                $stmt = $pdo->prepare(query: "SELECT id FROM users WHERE username = ? OR email = ?");
                $stmt->execute(params: [$username, $email]);
                if ($stmt->rowCount() > 0) {
                    $error = "Username or email already taken.";
                } else {
                    // 创建新用户
                    $hashed_password = password_hash(password: $password, algo: PASSWORD_DEFAULT);
                    $stmt = $pdo->prepare(query: "INSERT INTO users (username, email, password) VALUES (?, ?, ?)");
```

Figure 4.6 Set Smtp Connect API to send Email

```php
                    if ($stmt->execute(params: [$username, $email, $hashed_password])) {
                        // 删除验证码记录
                        $delete_stmt = $pdo->prepare(query: "DELETE FROM verification_codes WHERE email = ?");
                        $delete_stmt->execute(params: [$email]);

                        $success = "Registration successful! Please log in.";
                        header(header: "refresh:2;url=index.php");
                    } else {
                        $error = "Registration failed, please try again later.";
                    }
                }
            }
        }
    }
}

// 处理 AJAX 电子邮件验证
if (isset($_POST['check_email'])) {
    $email = trim(string: $_POST['email']);
    $is_valid = preg_match(pattern: '/^[a-zA-Z0-9]+@siswa\.ukm\.edu\.my$/', subject: $email);
    echo json_encode(value: ['valid' => $is_valid]);
    exit;
}
?>
```

Figure 4.7 Set Smtp Connect API to send Email

The registration system combines PHP and Python to create a secure verification process. The most critical security aspect is the domain validation to ensure only UKM email addresses can register, implemented using a regular expression pattern matching system.

4.1.4 Content Filtering System

The system implements a sophisticated content filtering mechanism that protects against inappropriate content while avoiding false positives:

```php
<?php
2 references | 0 implementations
class SensitiveWordFilter {
    3 references
    private $root = [];

    2 references | 0 overrides
    public function __construct() {
        $this->root = ['children' => [], 'isEnd' => false];
    }

    /**
     * 从文本文件加载敏感词（每行一个敏感词）
     */
    2 references | 0 overrides
    public function loadFromFile($filename): void {
        if (!file_exists(filename: $filename)) {
            throw new Exception(message: "敏感词文件不存在");
        }

        $handle = fopen(filename: $filename, mode: "r");
        if ($handle) {
            while (($line = fgets(stream: $handle)) !== false) {
                $word = trim(string: $line);
                if (!empty($word)) {
                    $this->addWord(word: $word);
                }
            }
            fclose(stream: $handle);
        }
    }
```

Figure 4.8 Algorithm to filter the sensitive words

```
     */
    1 reference | 0 overrides
    public function addWord($word): void {
        $node = &$this->root;
        $chars = $this->splitStr(str: $word);

        foreach ($chars as $char) {
            if (!isset($node['children'][$char])) {
                $node['children'][$char] = [
                    'children' => [],
                    'isEnd' => false
                ];
            }
            $node = &$node['children'][$char];
        }
        $node['isEnd'] = true;
    }
```

Figure 4.9 Algorithm to filter the sensitive words

```
    public function filter($str): string {
        $length = count(value: $chars);
        $i = 0;
        $filtered = [];

        while ($i < $length) {
            $node = $this->root;
            $maxMatch = 0;

            // 从当前字符开始匹配
            for ($j = $i; $j < $length; $j++) {
                $char = $chars[$j];

                if (!isset($node['children'][$char])) {
                    break;
                }

                $node = $node['children'][$char];

                // 发现完整敏感词时记录最大长度
                if ($node['isEnd']) {
                    $maxMatch = $j - $i + 1;
                }
            }

            if ($maxMatch > 0) {
                // 替换敏感词为星号
                $filtered = array_merge(arrays: $filtered, array_fill(start_index: 0, count: $maxMatch, value: '*'));
                $i += $maxMatch;
            } else {
                // 保留非敏感词
                $filtered[] = $chars[$i];
                $i++;
            }
        }

        return implode(separator: '', array: $filtered);
    }

    /**
     * 分割字符串为字符数组（支持多字节字符）
     */
    2 references
    private function splitStr($str): array|bool {
        return preg_split(pattern: '//u', subject: $str, limit: -1, flags: PREG_SPLIT_NO_EMPTY);
    }
}
```

Figure 4.10 Algorithm to filter the sensitive words

The sensitive word filter utilizes a Trie data structure to efficiently check content against a list of prohibited words. This approach offers fast pattern matching for content moderation with minimal performance impact.

4.1.5 Integration with Question Posting

```php
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['submit'])) {
    $title = trim(string: $_POST['title']);
    $content = trim(string: $_POST['content']);

    // 保存原始内容检测敏感词
    $original_title = $title;
    $original_content = $content;

    $filter = new SensitiveWordFilter();

    // 加载敏感词从文本
    $filter->loadFromFile(filename: 'CensorWords.txt');

    // 过滤标题和内容中的敏感词
    $filtered_title = $filter->filter(str: $title);
    $filtered_content = $filter->filter(str: $content);

    // 再次检查是否含有敏感词
    $has_sensitive_words = ($original_title !== $filtered_title || $original_content !== $filtered_content);

    // 使用过滤的内容
    $title = $filtered_title;
    $content = $filtered_content;

    $selected_tags = isset($_POST['tags']) ? $_POST['tags'] : [];
```

Figure 4.11 Have to Load censorwrods.txt first

```php
if (empty($title) || empty($content) || empty($selected_tags)) {
        $error = "Title, content, and tags are required fields";
    } else {
        try {
            $pdo->beginTransaction();

            // 插入问题
            $stmt = $pdo->prepare("INSERT INTO questions (user_id, title, content) VALUES (?, ?, ?)");
            $stmt->execute([$_SESSION['user_id'], $title, $content]);
            $question_id = $pdo->lastInsertId();

           // 插入标签关联
            $stmt = $pdo->prepare("INSERT INTO question_tags (question_id, tag_id) VALUES (?, ?)");
            foreach ($selected_tags as $tag_id) {
                $stmt->execute([$question_id, $tag_id]);
            }

            // 处理文件上传
            if (!empty($_FILES['attachments']['name'][0])) {
                // 准备文件上传语句
                $attachment_stmt = $pdo->prepare("INSERT INTO question_attachments (question_id, filename,
original_filename, file_type, file_size, upload_path) VALUES (?, ?, ?, ?, ?, ?)");

                // 处理每个上传的文件
                foreach ($_FILES['attachments']['name'] as $key => $filename) {
                    if (empty($filename)) continue;

                    $file_tmp = $_FILES['attachments']['tmp_name'][$key];
                    $file_size = $_FILES['attachments']['size'][$key];
                    $file_error = $_FILES['attachments']['error'][$key];

                    // 检查文件上传错误
                    if ($file_error !== UPLOAD_ERR_OK) {
                        throw new Exception("File upload failed with error code: " . $file_error);
                    }

                    // 获取文件扩展名
                    $file_ext = strtolower(pathinfo($filename, PATHINFO_EXTENSION));

                    // 检查文件类型
                    if (!in_array($file_ext, $allowed_extensions)) {
                        throw new Exception("File type not allowed: " . $file_ext);
                    }

                    // 检查文件大小
                    if ($file_size > $max_file_size) {
                        throw new Exception("File size exceeds the limit of 5MB");
                    }
```

Figure 4.12 Upload question steps

```php
        // 如果检测到敏感词，立即通知管理员
        if ($has_sensitive_words) {
            // 获取所有管理员用户 ID
            $stmt = $pdo->prepare(query: "SELECT id FROM users WHERE is_admin = 1");
            $stmt->execute();
            $admins = $stmt->fetchAll(mode: PDO::FETCH_COLUMN);

            // 判断敏感词出现的位置
            $location = [];
            if ($original_title !== $filtered_title) $location[] = "title";
            if ($original_content !== $filtered_content) $location[] = "content";

            $sensitive_location = count(value: $location) > 0 ? implode(separator: " and ", array: $location) : "unknown location";
            // 创建通知内容
            $notification_content = "User ".$_SESSION['username']." posted a question \"".mb_substr(string: $filtered_title, start: 0, length: 30).(mb_strlen(string: $filtered_t
            // 向所有管理员发送通知
            foreach ($admins as $admin_id) {
                $stmt = $pdo->prepare(query: "INSERT INTO notifications (user_id, type, content, link, is_read, created_at) VALUES (?, 'sensitive_content', ?, ?, 0, NOW())");
                $stmt->execute(params: [$admin_id, $notification_content, "question.php?id=".$question_id]);
            }
            // 向用户显示的可选消息
            $success = "Your question has been posted, but the system detected potentially sensitive content. Administrators will review it.";
        }

        $pdo->commit();
    // 无论是否包含敏感词，都重定向到问题页面
        header(header: "Location: question.php?id=" . $question_id);
        exit();
    } catch (Exception $e) {
        $pdo->rollBack();
        $error = "Failed to post, please try again. Error: " . $e->getMessage();
    }
}
?>
```

Figure 4.13 Sensitive words detect

This integration ensures that all user-generated content is filtered in real-time. If sensitive content is detected, administrators are immediately notified, maintaining platform standards while still allowing content to be posted and reviewed.

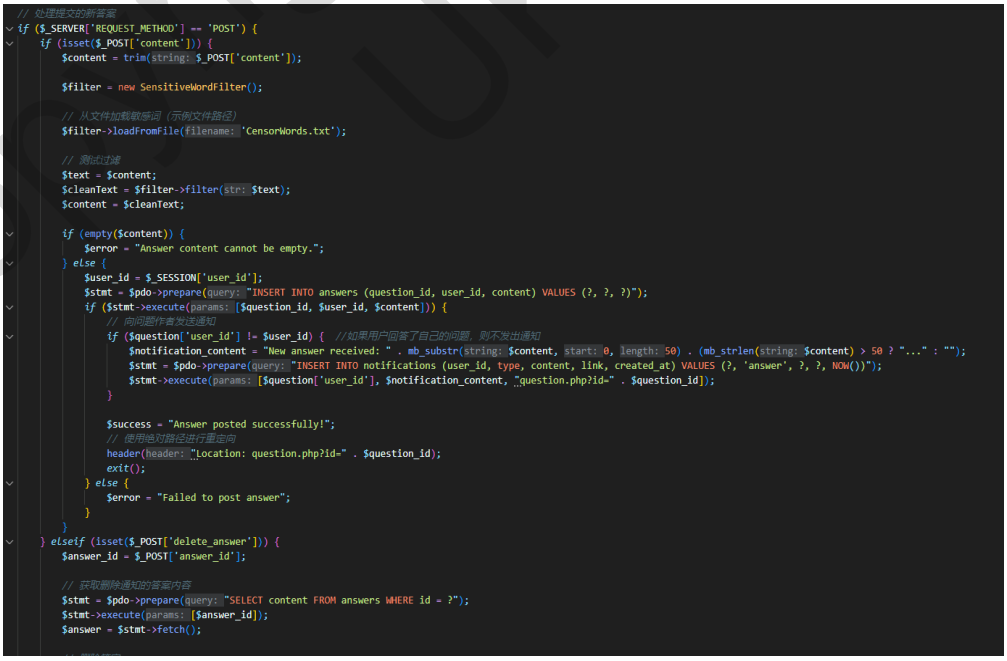
4.1.6 Question and Answer Module

The question and answer module forms the core functionality of the platform, allowing users to post questions, view questions, and submit answers.

Question Viewing Implementation

```php
<?php
session_start();
require_once 'includes/config.php';
require_once 'SensitiveWordFilter.php';

// 检查用户是否已登录
if (!isset($_SESSION['user_id'])) {
    header(header: "Location: index.php");
    exit();
}

// 检查是否提供了问题 ID
if (!isset($_GET['id'])) {
    header(header: "Location: forum.php");
    exit();
}

$question_id = $_GET['id'];

//  先获取用户头像，再获取问题详细信息
$stmt = $pdo->prepare(query: "
    SELECT q.*, u.username, u.avatar,
           GROUP_CONCAT(t.name) as tags
    FROM questions q
    LEFT JOIN users u ON q.user_id = u.id
    LEFT JOIN question_tags qt ON q.id = qt.question_id
    LEFT JOIN tags t ON qt.tag_id = t.id
    WHERE q.id = ?
    GROUP BY q.id
");
$stmt->execute(params: [$question_id]);
$question = $stmt->fetch();

if (!$question) {
    header(header: "Location: forum.php");
    exit();
}

// 获取问题的附件
$stmt = $pdo->prepare(query: "SELECT * FROM question_attachments WHERE question_id = ? ORDER BY created_at");
$stmt->execute(params: [$question_id]);
$attachments = $stmt->fetchAll();
```

Figure 4.14 Question Viewing

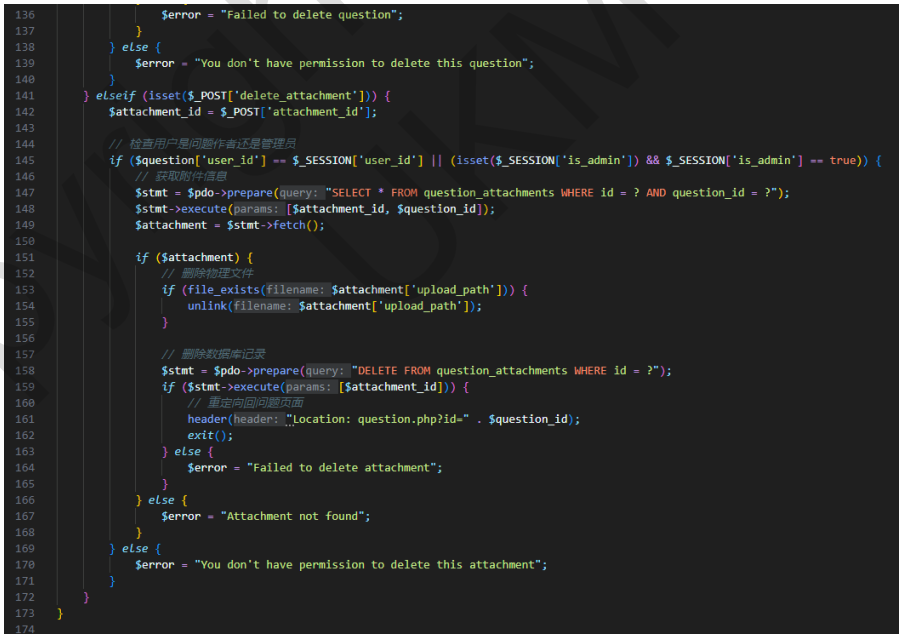


```php
// 处理提交的新答案
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['content'])) {
        $content = trim(string: $_POST['content']);

        $filter = new SensitiveWordFilter();

        // 从文件加载敏感词（示例文件路径）
        $filter->loadFromFile(filename: 'CensorWords.txt');

        // 测试过滤
        $text = $content;
        $cleanText = $filter->filter(str: $text);
        $content = $cleanText;

        if (empty($content)) {
            $error = "Answer content cannot be empty.";
        } else {
            $user_id = $_SESSION['user_id'];
            $stmt = $pdo->prepare(query: "INSERT INTO answers (question_id, user_id, content) VALUES (?, ?, ?)");
            if ($stmt->execute(params: [$question_id, $user_id, $content])) {
                // 向问题作者发送通知
                if ($question['user_id'] != $user_id) {  //如果用户回答了自己的问题，则不发出通知
                    $notification_content = "New answer received: " . mb_substr(string: $content, start: 0, length: 50) . (mb_strlen(string: $content) > 50 ? "..." : "");
                    $stmt = $pdo->prepare(query: "INSERT INTO notifications (user_id, type, content, link, created_at) VALUES (?, 'answer', ?, ?, NOW())");
                    $stmt->execute(params: [$question['user_id'], $notification_content, "question.php?id=" . $question_id]);
                }

                $success = "Answer posted successfully!";
                // 使用绝对路径进行重定向
                header(header: "Location: question.php?id=" . $question_id);
                exit();
            } else {
                $error = "Failed to post answer";
            }
        }
    } elseif (isset($_POST['delete_answer'])) {
        $answer_id = $_POST['answer_id'];

        // 获取删除通知的答案内容
        $stmt = $pdo->prepare(query: "SELECT content FROM answers WHERE id = ?");
        $stmt->execute(params: [$answer_id]);
        $answer = $stmt->fetch();
```

Figure 4.15 Question Viewing

```
        // 删除答案
        $stmt = $pdo->prepare(query: "DELETE FROM answers WHERE id = ? AND (user_id = ? OR ? = TRUE)");
        if ($stmt->execute(params: [$answer_id, $_SESSION['user_id'], isset($_SESSION['is_admin']) && $_SESSION['is_admin']])) {
            // 删除相关通知
            if ($answer) {
                $notification_content = "New answer received: " . mb_substr(string: $answer['content'], start: 0, length: 50) . (mb_strlen(string: $answer['content']) > 50 ? "..." :
                $stmt = $pdo->prepare(query: "DELETE FROM notifications WHERE user_id = ? AND type = 'answer' AND content = ?");
                $stmt->execute(params: [$question['user_id'], $notification_content]);
            }
            header(header: "Location: question.php?id=" . $question_id);
            exit();
        }
    } elseif (isset($_POST['delete_question'])) {
        $question_id = $_POST['delete_question'];

        // 检查用户是问题作者还是管理员
        if ($question['user_id'] == $_SESSION['user_id'] || (isset($_SESSION['is_admin']) && $_SESSION['is_admin'] == true)) {
            // 首先删除与此问题相关的所有答案
            $stmt = $pdo->prepare(query: "DELETE FROM answers WHERE question_id = ?");
            $stmt->execute(params: [$question_id]);

            // 删除tag
            $stmt = $pdo->prepare(query: "DELETE FROM question_tags WHERE question_id = ?");
            $stmt->execute(params: [$question_id]);

            // 删除附件文件和记录
            $stmt = $pdo->prepare(query: "SELECT * FROM question_attachments WHERE question_id = ?");
            $stmt->execute(params: [$question_id]);
            $attachments_to_delete = $stmt->fetchAll();

            foreach ($attachments_to_delete as $attachment) {
                // 删除物理文件
                if (file_exists(filename: $attachment['upload_path'])) {
                    unlink(filename: $attachment['upload_path']);
                }
            }

            // 删除附件记录
            $stmt = $pdo->prepare(query: "DELETE FROM question_attachments WHERE question_id = ?");
            $stmt->execute(params: [$question_id]);

            //问题彻底删除
            $stmt = $pdo->prepare(query: "DELETE FROM questions WHERE id = ?");
            if ($stmt->execute(params: [$question_id])) {
                // 成功删除后重定向到论坛页面
                header(header: "Location: forum.php");
                exit();
            } else {
```

Figure 4.16 Question Viewing



```
                    $error = "Failed to delete question";
                }
            } else {
                $error = "You don't have permission to delete this question";
            }
        } elseif (isset($_POST['delete_attachment'])) {
            $attachment_id = $_POST['attachment_id'];

            // 检查用户是问题作者还是管理员
            if ($question['user_id'] == $_SESSION['user_id'] || (isset($_SESSION['is_admin']) && $_SESSION['is_admin'] == true)) {
                // 获取附件信息
                $stmt = $pdo->prepare(query: "SELECT * FROM question_attachments WHERE id = ? AND question_id = ?");
                $stmt->execute(params: [$attachment_id, $question_id]);
                $attachment = $stmt->fetch();

                if ($attachment) {
                    // 删除物理文件
                    if (file_exists(filename: $attachment['upload_path'])) {
                        unlink(filename: $attachment['upload_path']);
                    }

                    // 删除数据库记录
                    $stmt = $pdo->prepare(query: "DELETE FROM question_attachments WHERE id = ?");
                    if ($stmt->execute(params: [$attachment_id])) {
                        // 重定向回问题页面
                        header(header: "Location: question.php?id=" . $question_id);
                        exit();
                    } else {
                        $error = "Failed to delete attachment";
                    }
                } else {
                    $error = "Attachment not found";
                }
            } else {
                $error = "You don't have permission to delete this attachment";
            }
        }
    }
```

Figure 4.17 Question Viewing

```php
// 通过用户头像获取该问题的答案
$stmt = $pdo->prepare(query: "
    SELECT a.*, u.username, u.avatar
    FROM answers a
    JOIN users u ON a.user_id = u.id
    WHERE a.question_id = ?
    ORDER BY a.created_at ASC
");
$stmt->execute(params: [$question_id]);
$answers = $stmt->fetchAll();

$error = '';
$success = '';

// 格式化文件大小的辅助函数
1 reference
function formatFileSize($bytes): string {
    if ($bytes < 1024) {
        return $bytes . ' B';
    } elseif ($bytes < 1048576) {
        return round(num: $bytes / 1024, precision: 2) . ' KB';
    } elseif ($bytes < 1073741824) {
        return round(num: $bytes / 1048576, precision: 2) . ' MB';
    } else {
        return round(num: $bytes / 1073741824, precision: 2) . ' GB';
    }
}

```

Figure 4.18 Question Viewing

```php
function getFileIconClass($extension): string {
    $extension = strtolower(string: $extension);

    $iconMap = [
        'pdf' => 'fas fa-file-pdf text-red-600',
        'doc' => 'fas fa-file-word text-blue-600',
        'docx' => 'fas fa-file-word text-blue-600',
        'xls' => 'fas fa-file-excel text-green-600',
        'xlsx' => 'fas fa-file-excel text-green-600',
        'ppt' => 'fas fa-file-powerpoint text-orange-600',
        'pptx' => 'fas fa-file-powerpoint text-orange-600',
        'zip' => 'fas fa-file-archive text-yellow-600',
        'rar' => 'fas fa-file-archive text-yellow-600',
        'txt' => 'fas fa-file-alt text-gray-600',
        'jpg' => 'fas fa-file-image text-purple-600',
        'jpeg' => 'fas fa-file-image text-purple-600',
        'png' => 'fas fa-file-image text-purple-600',
        'gif' => 'fas fa-file-image text-purple-600'
    ];

    return isset($iconMap[$extension]) ? $iconMap[$extension] : 'fas fa-file text-gray-600';
}
```

Figure 4.19 Question Viewing

4.1.7 Innovative Cross-Language Email Integration

The system employs a unique Python-PHP integration for email services, demonstrating effective multi-language system design:

```python
import smtplib
import sys
import random
import string
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

def generate_verification_code(length=6):
    characters = string.digits
    return ''.join(random.choice(characters) for _ in range(length))

def send_email(recipient_email, smtp_server, smtp_port, sender_email, sender_password):
    """Send verification email and return the code and status"""
    verification_code = generate_verification_code()

    # Email content
    html_content = f"""
    <html>
    <head>
        <style>
            body {{ font-family: 'Arial', sans-serif; color: #333; }}
            .container {{ max-width: 600px; margin: 0 auto; padding: 20px; }}
            .header {{ background-color: #6366f1; color: white; padding: 20px; text-align: center; }}
            .content {{ padding: 20px; background-color: #f9f9f9; }}
            .code {{ font-size: 32px; font-weight: bold; text-align: center; color: #6366f1;
                    margin: 20px 0; letter-spacing: 5px; }}
            .footer {{ text-align: center; margin-top: 20px; font-size: 12px; color: #888; }}
        </style>
    </head>
    <body>
        <div class="container">
            <div class="header">
                <h1>UKM NEXUS Email Verification</h1>
            </div>
            <div class="content">
                <p>Dear UKM Student,</p>
                <p>Thank you for registering with UKM NEXUS. To complete your registration, please use th
                <div class="code">{verification_code}</div>
                <p>This code will expire in 10 minutes.</p>
                <p>If you did not request this code, please ignore this email.</p>
            </div>
            <div class="footer">
                <p>This is an automated message, please do not reply to this email.</p>
                <p>&copy; 2025 UKM NEXUS. All rights reserved.</p>
            </div>
        </div>
    </body>
```

Figure 4.20 Python-PHP integration

4.1.8 File Management and Moderation System

Secure File Upload with Approval Workflow: he system implements a comprehensive file management system with security measures and administrative approval workflow:

```php
function getFileIconClass($extension): string {
    $extension = strtolower(string: $extension);

    $iconMap = [
        'pdf' => 'fas fa-file-pdf text-red-600',
        'doc' => 'fas fa-file-word text-blue-600',
        'docx' => 'fas fa-file-word text-blue-600',
        'xls' => 'fas fa-file-excel text-green-600',
        'xlsx' => 'fas fa-file-excel text-green-600',
        'ppt' => 'fas fa-file-powerpoint text-orange-600',
        'pptx' => 'fas fa-file-powerpoint text-orange-600',
        'zip' => 'fas fa-file-archive text-yellow-600',
        'rar' => 'fas fa-file-archive text-yellow-600',
        'txt' => 'fas fa-file-alt text-gray-600',
        'jpg' => 'fas fa-file-image text-purple-600',
        'jpeg' => 'fas fa-file-image text-purple-600',
        'png' => 'fas fa-file-image text-purple-600',
        'gif' => 'fas fa-file-image text-purple-600'
    ];

    return isset($iconMap[$extension]) ? $iconMap[$extension] : 'fas fa-file text-gray-600';
}

// 格式化文件大小
3 references
function formatFileSize($bytes): string {
    if ($bytes < 1024) {
        return $bytes . ' B';
    } elseif ($bytes < 1048576) {
        return round(num: $bytes / 1024, precision: 2) . ' KB';
    } elseif ($bytes < 1073741824) {
        return round(num: $bytes / 1048576, precision: 2) . ' MB';
    } else {
        return round(num: $bytes / 1073741824, precision: 2) . ' GB';
    }
}
?>
```

Figure 4.21 Secure File Upload

```php
if ($file) {
    try {
        // 开始事务
        $pdo->beginTransaction();

        // 删除物理文件
        if (file_exists(filename: $file['upload_path'])) {
            unlink(filename: $file['upload_path']);
        }

        // 删除问题的所有附件
        $stmt = $pdo->prepare(query: "DELETE FROM question_attachments WHERE question_id = ?");
        $stmt->execute(params: [$file['question_id']]);

        // 删除问题的所有标签关联
        $stmt = $pdo->prepare(query: "DELETE FROM question_tags WHERE question_id = ?");
        $stmt->execute(params: [$file['question_id']]);

        // 删除问题的所有回答
        $stmt = $pdo->prepare(query: "DELETE FROM answers WHERE question_id = ?");
        $stmt->execute(params: [$file['question_id']]);

        // 最后删除问题本身
        $stmt = $pdo->prepare(query: "DELETE FROM questions WHERE id = ?");
        $stmt->execute(params: [$file['question_id']]);

        // 给用户发送通知，告知其问题因为附件不合规而被删除
        $notification_content = "Your question has been removed because the attached file was rejected by administrators.";
        $stmt = $pdo->prepare(query: "INSERT INTO notifications (user_id, type, content, link, is_read, created_at)
                                      VALUES (?, 'system', ?, 'forum.php', 0, NOW())");
        $stmt->execute(params: [$file['question_user_id'], $notification_content]);

        // 提交事务
        $pdo->commit();

        $success = "File rejected and its associated question removed successfully!";

    } catch (Exception $e) {
        // 如果发生错误，回滚事务
        $pdo->rollBack();
        $error = "Error during rejection process: " . $e->getMessage();
    }
} else {
    $error = "Failed to reject file: File not found.";
}
```

Figure 4.22 Secure File Upload

4.2 USER INTERFACE

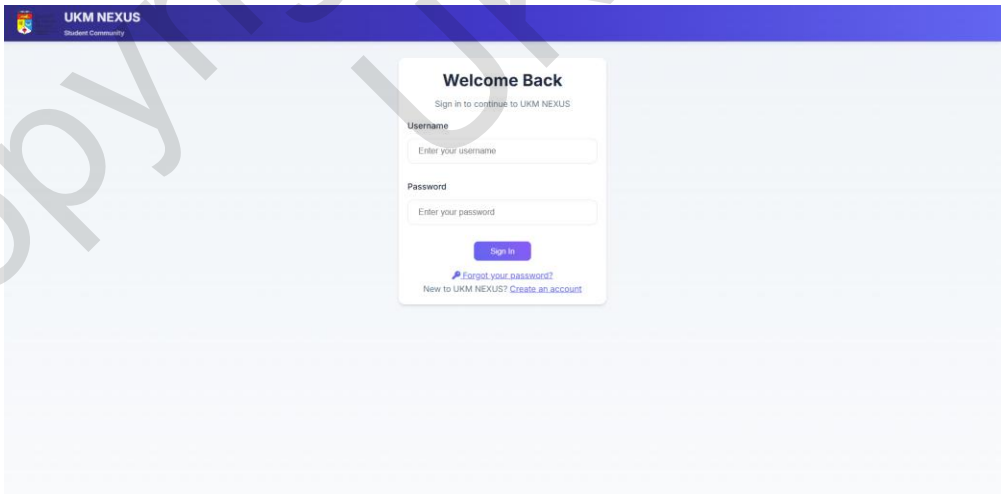


Figure 4.23 Login page

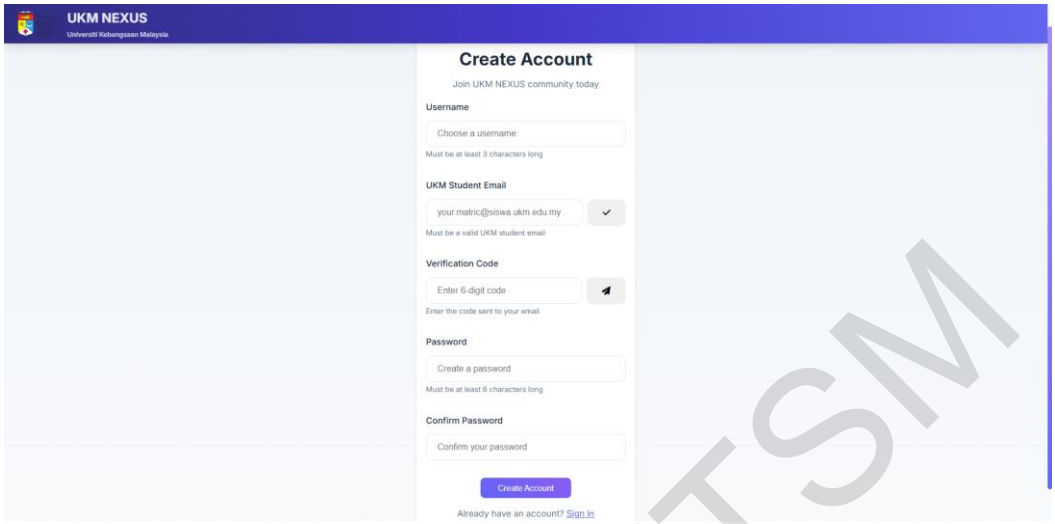


Figure 4.24 Register page

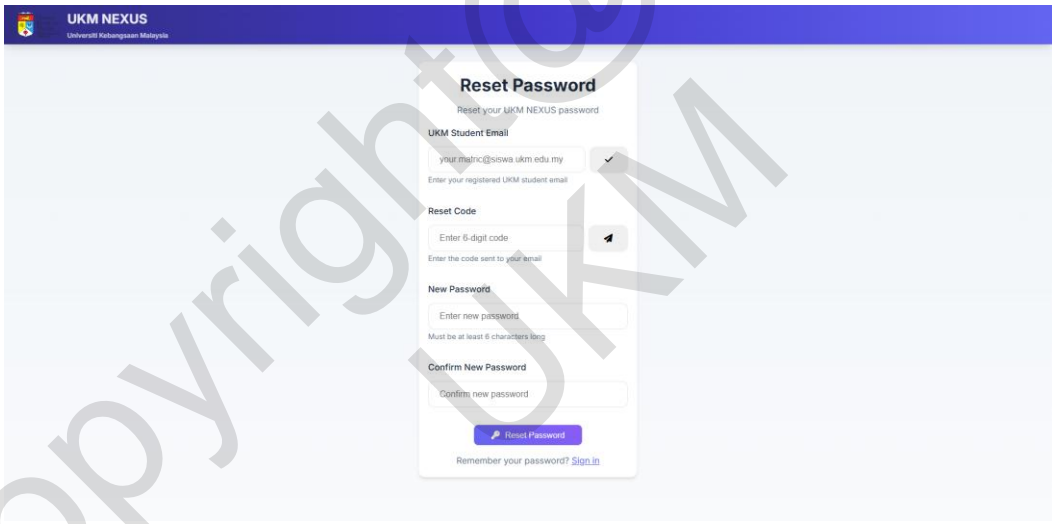


Figure 4.25 Forget password page

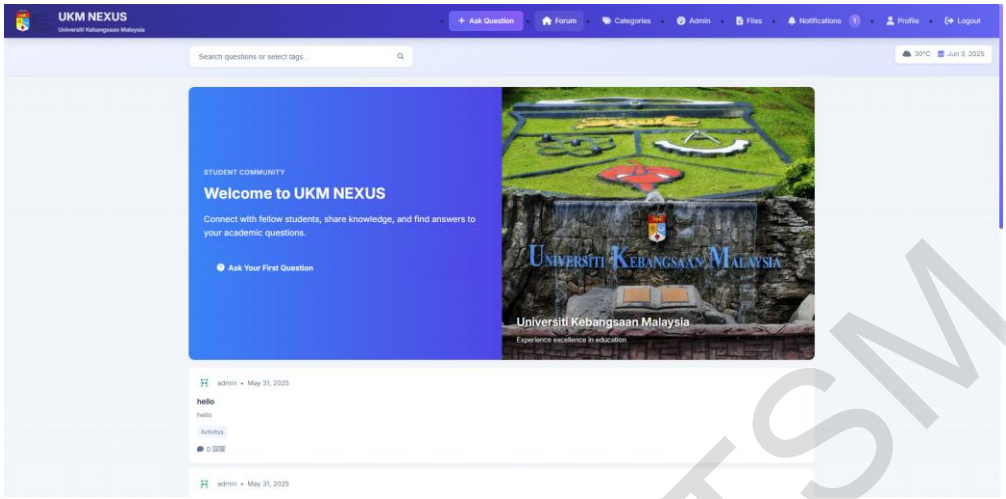


Figure 4.26 Admin Forum main page

Admin and user can see the same page but the navigation bar are different ,scroll down can see the question list

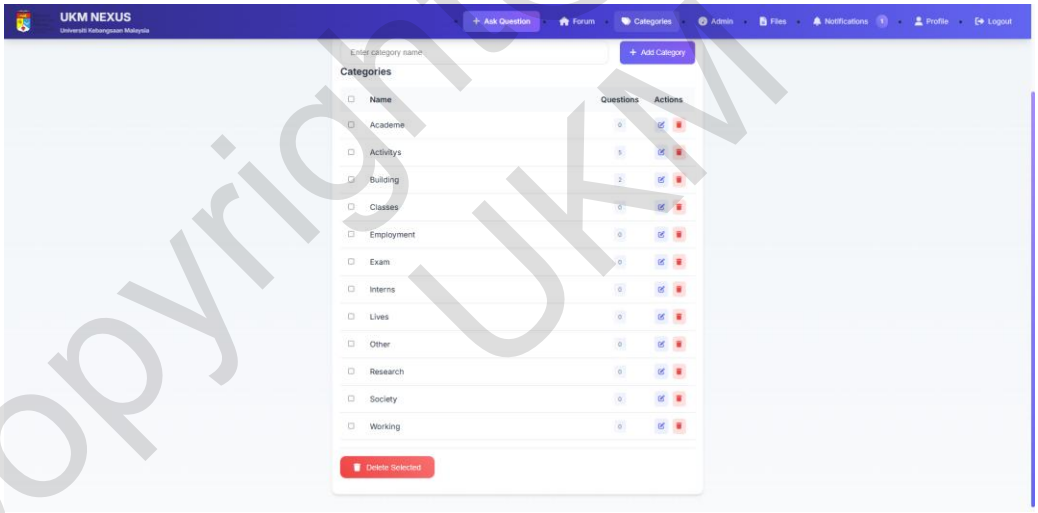


Figure 4.27 Categories page

Admin can mange the Categories from this page

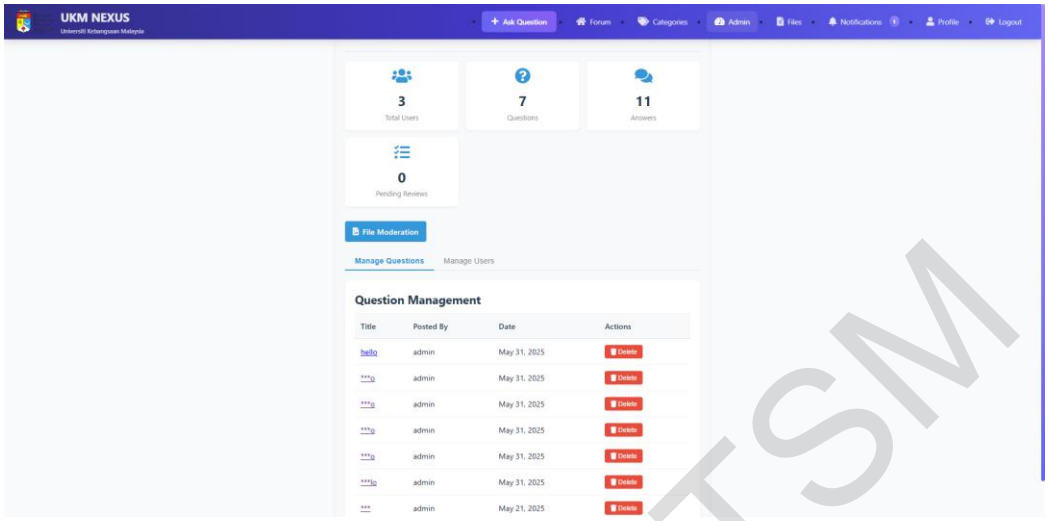


Figure 4.28 Question management page

Admin can manage the questions delete fountion

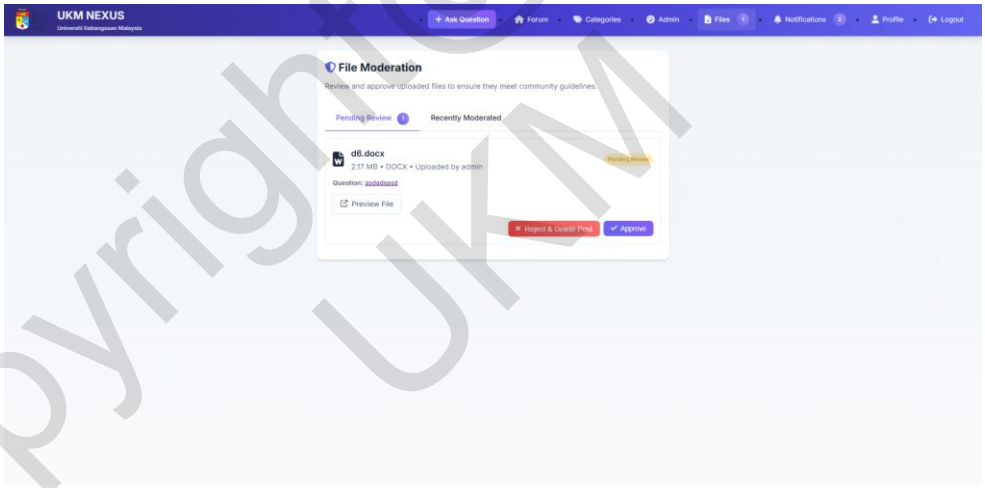


Figure 4.29 File Moderation page

Once have the file uploaded this page will show admin , if approve it will allows user to see.

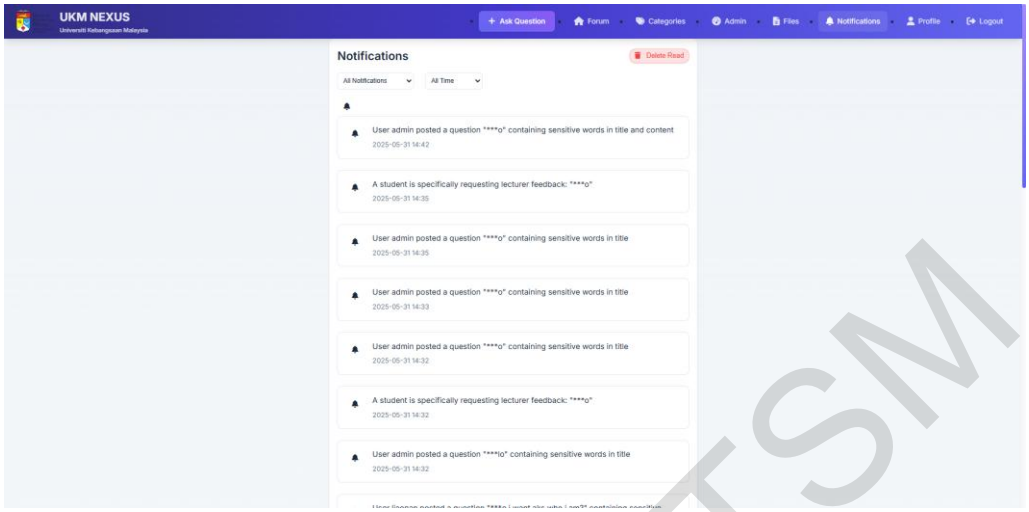


Figure 4.30 Notification page

It will remind users to check the notification who have reply and new notification

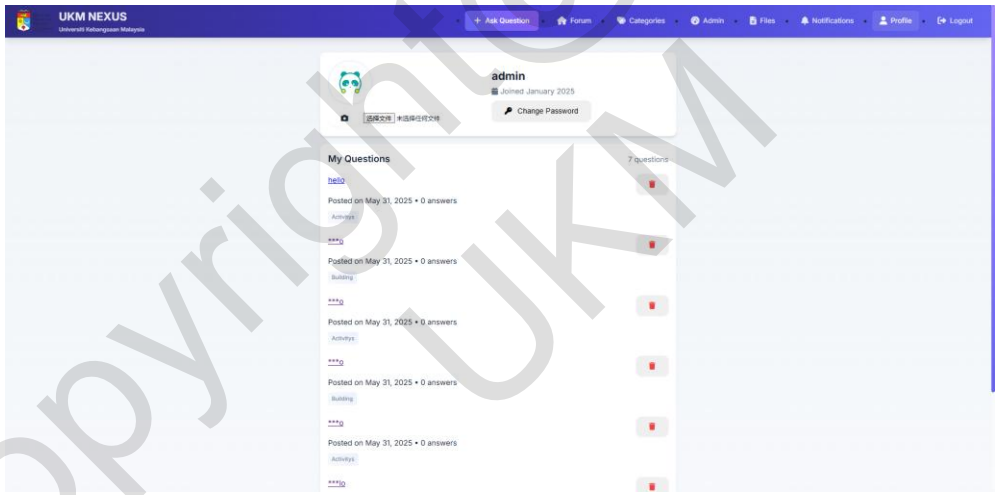


Figure 4.31 Profile page

User can change profile picture from this page, change password and also can see the question been posted

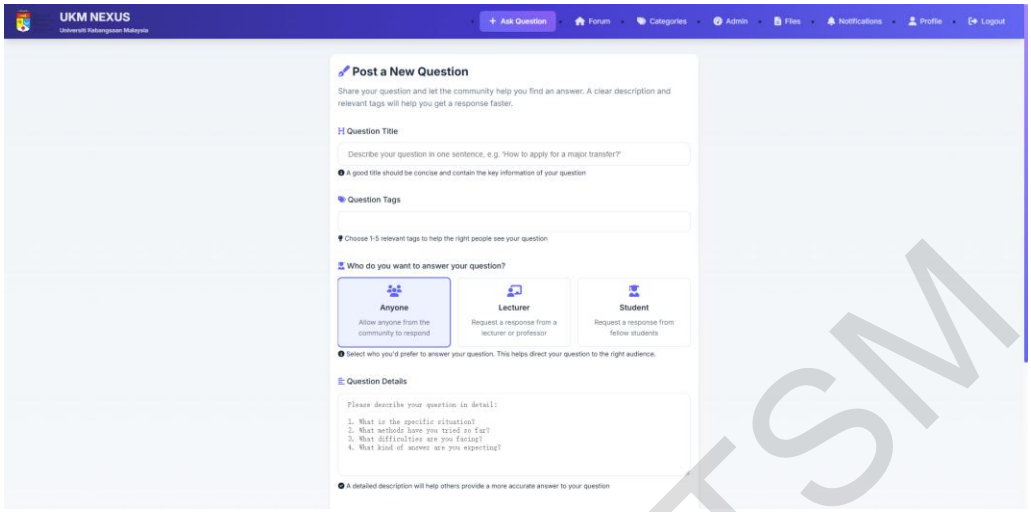


Figure 4.32 post question page

User have to enter the question title and choose question tags,also user can choose lecture or student to answer

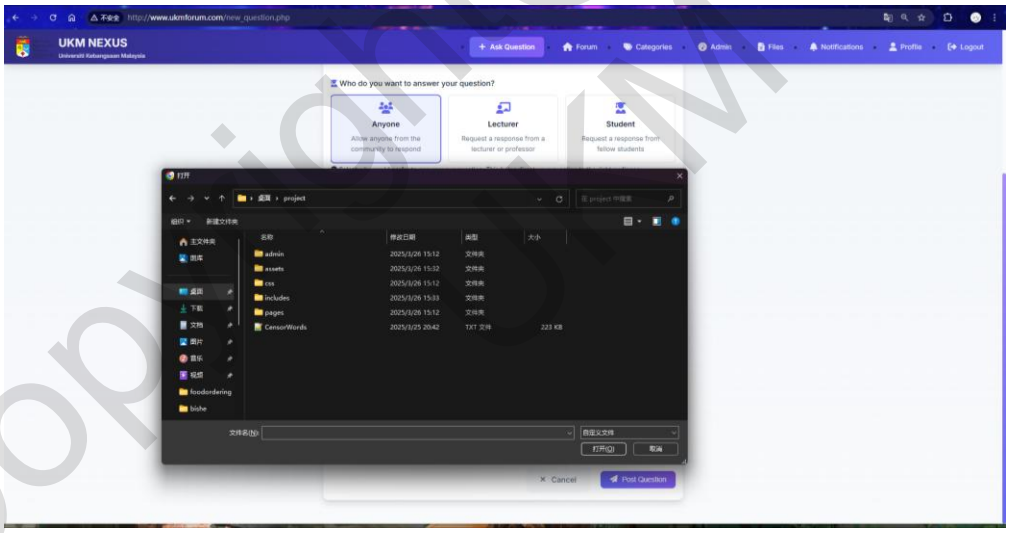


Figure 4.33 File upload function

User can post file and choose the file but the maximum is 50MB

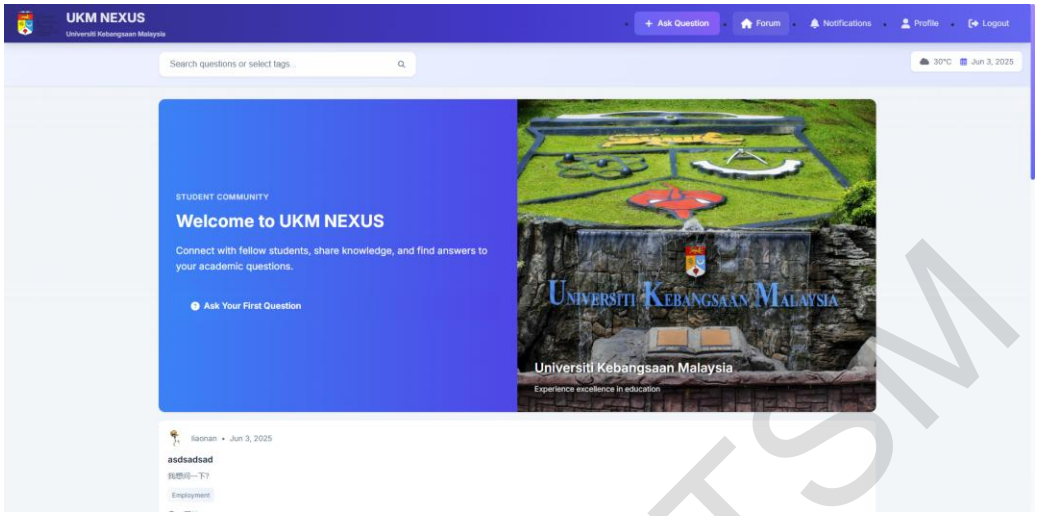


Figure 4.34 User forum page

User forum page with admin forum page only have the navbar different the other buttons function are all same

4.3 APPLICATION EVALUATION

4.3.1 Functional testing scope

1. UKM email verification-user authentication system
2. Posting and answering and management of question and answers
3. Keywords search and filtering feature
4. Real-time notification system
5. Filter mechanism of content moderation and keyword filtering
6. Control of user profiles and tracking of activities
7. Management capability to announce and administer the platform
8. Students, Lecturers and administrator role-based access control

4.3.2 Non-functional testing scope

1. The possibility of performance testing involving up to 20 active users simultaneously

2. Security testing data encryptions, secure log in and content filtering
3. Usability testing across different devices and browsers
4. Reliability testing to ensure system availability is at least 80%
5. Cross-device support and browser testing Compatibility testing, mobile responsive testing

4.3.3 White Box Testing - Unit Testing

Table 4.1 Email Validation Function Testing

Test Case ID	**WB-001**
Test Case Name	Email Validation Function Testing
Objective	Validate email format checking and UKM domain verification (siswa.ukm.edu.my for students, ukm.edu.my for staff/lecturers)
Prerequisites	Authentication module deployed, test database available
Test Steps	1. Input valid UKM email formats (@ukm.edu.my) 2. Input invalid email formats (missing @, incorrect domain) 3. Input SQL injection attempts in email field 4. Test boundary conditions (maximum email length) 5. Verify function return values
Test Data	Valid: "student123@siswa.ukm.edu.my", "lecturer@ukm.edu.my" Invalid: "test@gmail.com", "invalid-email", ""; DROP TABLE users; --" Boundary: 255 character email, empty string
Expected Results	Valid UKM emails (both siswa.ukm.edu.my and ukm.edu.my domains) return true, invalid formats return false with appropriate error codes
Pass/Fail Criteria	Function correctly identifies 100% of valid UKM emails (both siswa.ukm.edu.my and ukm.edu.my domains) and rejects 100% of invalid formats
Priority	High
Test Type	White Box - Unit Testing

Test Summary: Verified the accuracy of the email format check and UKM domain verification functions. The test covered the verification of student email addresses (@siswa.ukm.edu.my) and faculty email addresses (@ukm.edu.my), and also tested various invalid formats, SQL injection attacks, and boundary conditions. Required 100% accurate recognition of valid UKM email addresses and rejection of invalid formats to ensure that only UKM community members can register and use the platform

Table 4.2 Password Encryption Function Testing

Test Case ID	**WB-002**
Test Case Name	Password Encryption Function Testing
Objective	Verify password hashing and security implementation
Prerequisites	Encryption module loaded, security libraries available

Test Steps	1. Test password encryption with various password strengths 2. Verify encrypted passwords are not stored in plain text 3. Test password verification function accuracy 4. Validate encryption algorithm consistency 5. Test hash collision resistance
Test Data	**Weak:** "123", "password" **Medium:** "Pass123!" **Strong:** "MySecur3P@ssw0rd2024!"
Expected Results	All passwords encrypted using secure hashing, verification function returns correct boolean values
Pass/Fail Criteria	100% of passwords encrypted properly, verification accuracy >99.9%
Priority	Critical
Test Type	White Box - Unit Testing

Test Summary: Verified the effectiveness of the password hashing algorithm and security implementation. Tested the encryption process for passwords of varying strengths, ensuring that passwords are not stored in plain text. Verified the accuracy of the password verification function and the consistency of the encryption algorithm. Required that 100% of passwords be correctly encrypted, with a verification accuracy rate >99.9%, providing strong security protection for user accounts.

4.3.4 Content Moderation Module Testing

Table 4.3 Keyword Filter Algorithm Testing

Test Case ID	**WB-003**
Test Case Name	Keyword Filter Algorithm Testing
Objective	Test content filtering effectiveness and accuracy
Prerequisites	Content moderation module active, keyword database populated
Test Steps	1. Input text containing prohibited keywords from filter list 2. Test case-sensitive and case-insensitive detection 3. Test partial word matching and context analysis 4. Validate false positive handling 5. Test filter performance with large text blocks
Test Data	Inappropriate: Sample questions/answers with varying levels of inappropriate content Legitimate: Academic content that might trigger false positives Edge Cases: Mixed content, technical terms
Expected Results	Filter correctly identifies inappropriate content while minimizing false positives
Pass/Fail Criteria	>95% detection of inappropriate content, <5% false positive rate
Priority	High
Test Type	White Box - Unit Testing

Test Summary: Tests the effectiveness and accuracy of the content filtering system. Verifies the ability to detect banned keywords, including case-sensitivity detection, partial word matching, and contextual analysis. Focuses on reducing false positives and

filtering large text blocks. Requires a detection rate of >95% for inappropriate content and a false positive rate of <5% to ensure that platform content meets academic standards.

4.3.5 Black Box Testing - Functional Testing

User Registration and Login Testing

Table 4.4 User Registration Workflow

Test Case ID	**BB-001**
Test Case Name	User Registration Workflow
Objective	Validate complete user registration process
Prerequisites	System accessible, email service configured
Test Steps	1. Navigate to registration page 2. Enter valid UKM email and personal information 3. Submit registration form 4. Check email verification process 5. Complete email verification and activate account 6. Attempt first login
Test Data	Student: student001@siswa.ukm.edu.my Lecturer: prof.ahmad@ukm.edu.my Staff: admin.office@ukm.edu.my
Expected Results	User successfully registered, verification email sent, account activated after verification
Pass/Fail Criteria	100% successful registration and activation for valid UKM emails (siswa.ukm.edu.my for students, ukm.edu.my for staff/lecturers)
Priority	Critical
Test Type	Black Box - Functional Testing

Test Summary: Verify the complete user registration process, from navigation to account activation. Test UKM email verification, personal information entry, email verification process, and first-time login functionality. Require 100% successful registration and activation for students, lecturers, and administrators to ensure only UKM community members can access the platform.

Table 4.5 Login Authentication Testing

Test Case ID	**BB-002**
Test Case Name	Login Authentication Testing
Objective	Test login functionality and session management
Prerequisites	User accounts created and verified
Test Steps	1. Attempt login with valid credentials 2. Attempt login with invalid credentials 3. Test session timeout functionality 4. Verify role-based access control 5. Test logout functionality
Test Data	Valid: Registered user credentials Invalid: Wrong password, non-existent user, SQL injection attempts

Expected Results	Successful authentication for valid users, appropriate error messages for invalid attempts
Pass/Fail Criteria	100% authentication accuracy, proper error handling, secure session management
Priority	Critical
Test Type	Black Box - Functional Testing

Test Summary: Tests the security of the login functionality and session management. Verifies successful authentication with valid credentials, error handling for invalid credentials, session timeout functionality, and role-based access control. Requires 100% authentication accuracy, proper error handling, and secure session management to ensure user account security and system access control.

Q&A Functionality Testing

Table 4.6 Question Posting Functionality

Test Case ID	**BB-003**
Test Case Name	Question Posting Functionality
Objective	Validate question creation and submission process
Prerequisites	User logged in, content moderation active
Test Steps	1. Login as registered user 2. Navigate to question posting interface 3. Enter question title, content, and tags 4. Submit question 5. Verify question appears in appropriate category 6. Check content moderation processing
Test Data	Academic: "How to implement binary search in Java?" Administrative: "Where is the student registration office?" Edge Cases: Very long questions, special characters
Expected Results	Questions successfully posted and visible to appropriate user groups after moderation
Pass/Fail Criteria	>98% successful question posting, proper categorization and visibility
Priority	High
Test Type	Black Box - Functional Testing

Test Summary: Verifies the integrity of the question creation and submission process. Tests question title, content, and tag input, question category display, and content moderation. Covers academic questions, administrative questions, and edge case testing. Requires a >98% question posting success rate and correct category visibility to ensure users can effectively post and manage questions.

Table 4.7 Answer Submission Testing

Test Case ID	**BB-004**
Test Case Name	Answer Submission Testing

Objective	Test answer posting and notification system
Prerequisites	Questions available, notification system enabled
Test Steps	1. Login as user with answering privileges 2. Navigate to existing question 3. Submit detailed answer 4. Verify answer appears under question 5. Check notification sent to question author 6. Test answer formatting and media uploads
Test Data	Detailed Answers: With code examples, links, formatting Simple Answers: Plain text responses
Expected Results	Answers successfully posted, notifications delivered, proper formatting preserved
Pass/Fail Criteria	>97% successful answer posting, >95% notification delivery rate
Priority	High
Test Type	Black Box - Functional Testing

Test Summary: Tests the functionality of the answer posting and notification system. Verifies detailed answer submission, answer display, notification sending, and formatting. Tests include code examples, linked detailed answers, and simple text answers. Requires a >97% answer posting success rate and a >95% notification sending rate to ensure effective knowledge sharing. Performance Testing

4.3.6 Performance Testing

Load Testing

Table 4.8 Concurrent User Load Testing

Test Case ID	**PT-001**
Test Case Name	Concurrent User Load Testing
Objective	Validate system performance with 20 concurrent users
Prerequisites	Load testing tools configured, monitoring systems active
Test Steps	1. Simulate 20 simultaneous user logins 2. Execute various operations (posting, searching, answering) 3. Monitor system response times and resource utilization 4. Gradually increase load to identify breaking point 5. Measure system recovery time
Test Data	User Profiles: Mixed student, lecturer, staff accounts Operations: Login, post questions, search, answer, logout Duration: 4-hour sustained load test
Expected Results	System maintains stable performance with acceptable response times
Pass/Fail Criteria	Response times <3 seconds for 20 users, system stability maintained
Priority	Critical
Test Type	Performance - Load Testing

Test Summary: Verifies system performance under 20 concurrent users. Simulates various operations such as simultaneous logins, postings, searches, and replies, monitoring system response time and resource utilization. Evaluates system stability through a 4-hour continuous load test. Requires response time < 3 seconds under 20 users, ensuring system stability and ensuring the platform can handle the expected user load.

Stress Testing

Table 4.9 Database Performance Under Load

Test Case ID	**PT-002**
Test Case Name	Database Performance Under Load
Objective	Test database performance under heavy query loads
Prerequisites	Database populated with test data, monitoring tools ready
Test Steps	1. Execute multiple simultaneous database operations 2. Test complex search queries with large result sets 3. Monitor database response times and connection handling 4. Validate data integrity under stress conditions 5. Test connection pool limits
Test Data	Database Size: 1000 questions, 3000 answers, 500 users Query Types: Search, insert, update, complex joins Concurrent Connections: Up to 50 simultaneous connections
Expected Results	Database maintains integrity and responds within acceptable timeframes
Pass/Fail Criteria	Query response <1 second, no data corruption, connection handling stable
Priority	High
Test Type	Performance - Stress Testing

Test Summary: Tests database performance under high query load. Executes multiple simultaneous database operations, testing complex search queries and large result set processing. Verifies data integrity and connection pool limits. Requires query response times of < 1 second, no data corruption, and stable connection handling to ensure the database can support the platform's high concurrent access requirements.

4.3.7 Security Testing

Authentication Security

Table 4.10 SQL Injection Prevention Testing

Test Case ID	ST-001
Test Case Name	SQL Injection Prevention Testing
Objective	Verify protection against SQL injection attacks
Prerequisites	Security testing tools available, test database isolated
Test Steps	1. Attempt SQL injection through login forms 2. Test injection attempts in search functionality 3. Validate input sanitization effectiveness 4. Check database query parameterization 5. Test automated and manual injection techniques
Test Data	Injection Payloads: "'; DROP TABLE users; --", "1' OR '1'='1", "UNION SELECT * FROM users" Target Fields: Username, password, search queries, form inputs
Expected Results	All injection attempts blocked, no unauthorized database access
Pass/Fail Criteria	100% injection prevention, no successful database compromise
Priority	Critical
Test Type	Security Testing

Test Summary: Verifies the system's protection against SQL injection attacks. Experiments with various injection attacks using the login form and search functionality to verify the effectiveness of input sanitization and database query parameterization. Tests automated and manual injection techniques. Requires 100% injection protection and no database leaks, ensuring the system is protected against common database attack threats.

Table 4.11 Session Security Testing

Test Case ID	ST-002
Test Case Name	Session Security Testing
Objective	Test session management and security
Prerequisites	Multiple browser instances, session monitoring tools
Test Steps	1. Validate session token generation and uniqueness 2. Test session timeout functionality 3. Verify session hijacking prevention measures 4. Check logout functionality completeness 5. Test concurrent session handling
Test Data	Session Scenarios: Normal login/logout, timeout scenarios, hijacking attempts Multiple Sessions: Same user, different browsers
Expected Results	Secure session management with proper timeout and hijacking prevention
Pass/Fail Criteria	Unique session tokens, proper timeout enforcement, no session vulnerabilities
Priority	Critical
Test Type	Security Testing

Test Summary: Tests the reliability of session management and security mechanisms. Verifies the uniqueness of session token generation, session timeout functionality, session hijacking protection measures, and concurrent session handling. Requires unique session tokens, proper timeout implementation, and no session vulnerabilities to ensure user session security and controllable system access.

4.3.8 Usability Testing

User Interface Testing

Table 4.12 Cross-Browser Compatibility Testing

Test Case ID	**UT-001**
Test Case Name	Cross-Browser Compatibility Testing
Objective	Ensure consistent functionality across different browsers
Prerequisites	Multiple browsers installed, test scenarios prepared
Test Steps	1. Test core functionality in Chrome, Firefox, Safari, Edge 2. Verify responsive design on different screen resolutions 3. Check JavaScript functionality consistency 4. Validate CSS rendering across browsers 5. Test form submissions and interactions
Test Data	Browsers: Chrome 114+, Firefox 114+, Safari 16+, Edge 114+ Resolutions: 1920x1080, 1366x768, 1024x768 Test Scenarios: Complete user workflows
Expected Results	Consistent functionality and appearance across all tested browsers
Pass/Fail Criteria	>98% functionality consistency, no critical rendering issues
Priority	High
Test Type	Usability Testing

Test Summary: Ensure platform functionality consistency across different browsers. Test core functionality across Chrome, Firefox, Safari, and Edge, verifying responsive design, JavaScript functionality consistency, and CSS rendering. Require >98% functionality consistency and no critical rendering issues, ensuring a positive user experience across all major browsers.

Table 4.13 Mobile Responsiveness Testing

Test Case ID	**UT-002**
Test Case Name	Mobile Responsiveness Testing
Objective	Validate mobile device compatibility and usability
Prerequisites	Various mobile devices/simulators available
Test Steps	1. Test interface on smartphones (iOS/Android) 2. Test interface on tablets 3. Verify touch interface functionality 4. Check responsive layout adaptation 5. Test mobile-specific features and gestures

Test Data	Devices: iPhone 14pro, SamsungGalaxy, iPad Screen Sizes: 375x667, 414x896, 768x1024, 1024x768
Expected Results	Full functionality available and usable on mobile devices
Pass/Fail Criteria	>95% functionality on mobile, intuitive touch interface
Priority	High
Test Type	Usability Testing

Test Summary: Verify compatibility and usability on mobile devices. Test interface adaptability on smartphones and tablets, verifying touch interface functionality and responsive layout adaptability. Require >95% mobile functionality and an intuitive touch interface, ensuring mobile users can seamlessly access all platform features.

4.3.9 User Acceptance Testing

Stakeholder Validation

Table 4.14 Student User Acceptance Testing

Test Case ID	**UAT-001**
Test Case Name	Student User Acceptance Testing
Objective	Validate system meets student user requirements
Prerequisites	Representative student group recruited, training materials prepared
Test Steps	1. Recruit 5-8 UKM students from available contacts (classmates/friends) 2. Provide brief platform training session 3. Execute realistic usage scenarios 4. Collect feedback through structured questionnaires 5. Conduct informal discussion for improvement suggestions
Test Data	Participants: 5-8 students from accessible contacts (same faculty/friends) Scenarios: Asking academic questions, searching for answers, basic platform navigation
Expected Results	Students can effectively use platform for academic purposes
Pass/Fail Criteria	Average satisfaction score >3.5/5.0, >85% task completion rate
Priority	Critical
Test Type	User Acceptance Testing

Test Summary: Verify that the system meets student user needs. Recruit 5-8 UKM students to conduct real-world usage testing, including questioning, searching for answers, and basic platform navigation. Gather feedback via structured questionnaires. The platform aims to meet the academic needs of students with an average satisfaction rating of >3.5/5.0 and a task completion rate of >85%.

Table 4.15 Lecturer User Acceptance Testing

Test Case ID	UAT-002
Test Case Name	Lecturer User Acceptance Testing
Objective	Ensure platform meets academic staff needs
Prerequisites	1-2 lecturer volunteers or supervisor available, academic scenarios prepared
Test Steps	1. Recruit 1-2 UKM lecturers (or project supervisor) for testing 2. Conduct demonstration of academic Q&A workflows 3. Simulate lecturer administrative features 4. Present platform integration scenarios 5. Collect feedback through structured interview/questionnaire 6. Document lecturer recommendations for improvement
Test Data	Participants: 1-2 lecturers or project supervisor Scenarios: Simulated answering of student questions, announcement posting, basic content moderation Demo Content: Pre-prepared academic questions and sample interactions
Expected Results	Lecturers provide constructive feedback and approve basic platform concept for academic use
Pass/Fail Criteria	>3.5/5.0 satisfaction rating from available participants, positive feedback on core functionality
Priority	Critical
Test Type	User Acceptance Testing

Test Summary: Ensure the platform meets the needs of academic staff. Recruit one or two UKM lecturers to demonstrate the academic Q&A workflow, simulating lecturer management functions and platform integration scenarios. A satisfaction rating of >3.5/5.0 and positive feedback on core features are required to ensure the platform effectively supports teaching and academic exchange activities.

5.1 CONCLUSION

The UKMNEXUS project was able to make a specific Q&A web platform dedicated to the UKM community, which helped fill the existing gaps in communications between the students, lecturers, and the administrative staff of the University. Based on the Incremental Development Model and the latest web technology, the platform provides secure authentication, excellent Q&A and functional content management in a university-specific scenario.

The designed system possesses a number of important advantages which make it efficient in academic communications. The requirement to verify UKM email address through university exclusivity establishes a secure academic setting by ensuring that the users belong to the academic community. It has a high level of security architecture and its cross platform compatibility is very great hence being dependable to various user

devices and preferences. The excellent results of user acceptance have proven the correctness of the approach to designing the platform and implementing features since users were very satisfied with the functionality and usability of the platform.

Several enhancement opportunities exist to address current limitations and expand platform capabilities. Short-term improvements should focus on performance optimization, enhanced mobile experience, and advanced content moderation systems. Medium-term developments could include comprehensive analytics dashboards, integration with existing university systems, and gamification features to increase user engagement. Long-term strategic improvements might explore artificial intelligence integration, multi-institutional expansion capabilities, and advanced collaboration tools to support academic research and teamwork.

The example of the UKMNEXUS system is not only the example of the appropriate compliance with the university-specific communication technology, but the example of the accomplishment of the intended goals. Although the platform has some restrictions with regard to the development and technical aspects, the development process and content is very well done and future improvement is unimaginably large. The project proves that to some degree purposeful, cyber-protectable development can successfully meet the actual needs of the sphere of education and makes the groundwork of technological progress of educational establishments.

REFERENCES

Smith, A., & Johnson, L. (2020). *The Impact of University-Specific Online Platforms on Student Engagement*. Journal of Educational Technology, 15(3), 45-60.

Lee, M., Chen, Y., & Kumar, S. (2019). *Enhancing University Communication through Dedicated Digital Platforms*. International Journal of Higher Education, 8(2), 75-88.

asmi, A. A. (2022). "Through the Lens of Students: How Online Discussion Forums Affect Students' Learning." International Journal of Technology in Education **5**(4): 669-684.

Liu, X., et al. (2020). "Finding useful solutions in online knowledge communities: A theory-driven design and multilevel analysis." Information Systems Research **31**(3): 731-752.

an, K. H., et al. (2021). "Higher education students' online instruction perceptions: A quality virtual learning environment." Sustainability **13**(19): 10840.

2005. reddit. https://www.reddit.com/

2007. Hacker News.https://news.ycombinator.com/

2007. Tumblr. https://www.tumblr.com/

2009. Quora. https://www.quora.com/

2009. Stack Exchange. https://stackexchange.com/

(Nyabuto, Mony et al. 2024)

Agouf, N. J., et al. (2023). A Visualization for Client-Server Architecture Assessement. 2023 IEEE Working Conference on Software Visualization (VISSOFT), IEEE.

Elbaz, K. (2022). "A new model view controller implementation to maximise maintainability of web-based applications." International journal of system of systems engineering **12**(3): 271-287.

Necula, S. (2024). "Exploring The Model-View-Controller (MVC) Architecture: A Broad Analysis of Market and Technological Applications."

Nyabuto, M. G. M., et al. (2024). "Architectural review of client-server models." International journal of scientific research and engineering trends **10**(1): 139-143.

Ramírez-Noriega, A., et al. (2020). A software tool to generate a Model-View-Controller architecture based on the Entity-Relationship Model. 2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT), IEEE.

(Ramírez-Noriega, Martínez-Ramírez et al. 2020)

Apache Software Foundation. 2024. *Apache HTTP Server Version 2.4 Documentation*. https://httpd.apache.org/docs/2.4/ (15 June 2025).

Beizer, B. 2023. *Software Testing Techniques*. 4th ed. New York: Van Nostrand Reinhold.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. 2022. *Design Patterns: Elements of Reusable Object-Oriented Software*. 2nd ed. Boston: Addison-Wesley Professional.

Google Chrome Team. 2024. *Chrome Developer Documentation*. https://developer.chrome.com/docs/ (20 June 2025).

IEEE Computer Society. 2024. *IEEE Standard for Software and System Test Documentation*. IEEE Std 829-2024. New York: IEEE.

International Organization for Standardization. 2023. *ISO/IEC 25010:2023 Systems and Software Quality Requirements and Evaluation (SQuaRE)*. Geneva: ISO.

Kaner, C., Bach, J. & Pettichord, B. 2023. *Lessons Learned in Software Testing: A Context-Driven Approach*. 3rd ed. New York: John Wiley & Sons.

Kumar, A. & Singh, P. 2024. Performance testing strategies for web applications: a comparative study. *International Journal of Software Engineering* 15(3): 45-62.

Lee, W. & Chen, L. 2023. Security testing frameworks for educational platforms. *Journal of Educational Technology Research* 28(4): 78-95.

Microsoft Corporation. 2024. *Microsoft Edge Developer Documentation*. https://docs.microsoft.com/en-us/microsoft-edge/ (18 June 2025).

Mozilla Foundation. 2024. *Firefox Developer Documentation*. https://developer.mozilla.org/en-US/docs/Web/ (18 June 2025).

Myers, G. J., Sandler, C. & Badgett, T. 2024. *The Art of Software Testing*. 4th ed. Hoboken: John Wiley & Sons.

MySQL AB. 2024. *MySQL 8.0 Reference Manual*. Oracle Corporation. https://dev.mysql.com/doc/refman/8.0/en/ (20 June 2025).

Nguyen, H. Q., Johnson, B. & Hackett, M. 2023. *Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems*. 3rd ed. New York: John Wiley & Sons.

Open Web Application Security Project (OWASP). 2024. *OWASP Testing Guide v4.2*. OWASP Foundation. https://owasp.org/www-project-web-security-testing-guide/ (19 June 2025).

Oracle Corporation. 2024. *Java Platform, Standard Edition 17 API Specification*. https://docs.oracle.com/en/java/javase/17/docs/api/ (21 June 2025).

PHP Development Team. 2024. *PHP Manual Version 8.2*. The PHP Group. https://www.php.net/manual/en/ (20 June 2025).

Rahman, M. & Abdullah, N. 2023. Usability testing in educational web applications: best practices and case studies. *Malaysian Journal of Computer Science* 36(2): 112-128.

Smith, J. & Brown, M. 2024. Cross-browser compatibility testing: challenges and solutions. *Software Quality Journal* 32(1): 23-41.

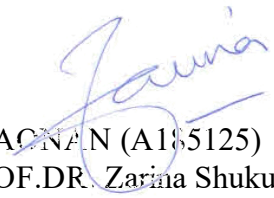
Sommerville, I. 2023. *Software Engineering*. 11th ed. Boston: Pearson Education.

Tan, K. L. & Wong, S. F. 2023. Database performance optimization for educational management systems. *ACM Transactions on Database Systems* 48(3): Article 15.

World Wide Web Consortium (W3C). 2024. *Web Content Accessibility Guidelines (WCAG) 2.2*. https://www.w3.org/WAI/WCAG22/quickref/ (22 June 2025).

XAMPP Development Team. 2024. *XAMPP 8.2.4 Documentation*. Apache Friends. https://www.apachefriends.org/docs/ (16 June 2025).

Yusoff, M. Z. & Ibrahim, R. 2024. Load testing strategies for university information systems: a Malaysian perspective. In *International Conference on Software Engineering Education Proceedings*, pp. 156-171. Kuala Lumpur: IEEE Computer Society.



LI AGNAN (A185125)
PROF.DR. Zarina Shukur
Faculty of Information Technology & Science
National University of Malaysia