🎓 NamiQuiz - Namibian Learning Quiz System

Complete Technical Documentation & Presentation

Course: Web Application Development (WAD621S)
Lecturer: MS. J. MUNTUUMO
Submission Date: 19 October 2025

Group Members

- **Natasha Chilinda** – 221143122
- **Gehas Haimbili** – 223039128
- **Maria Armas** – 223109630

📑 **Table of Contents**

---

1. Executive Summary

NamiQuiz is a web-based interactive learning platform designed to enhance education in Namibia through engaging quizzes covering Namibian History, Culture, Computer Science, General Knowledge, and Mathematics. The system provides:

- **110+ questions** across 5 categories
- **Real-time feedback** and scoring
- **User authentication** and progress tracking
- **Responsive design** for desktop and mobile
- **Password recovery** functionality

Key Technologies:

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** PHP 8.0

- **Database:** MySQL 8.0
- **Server:** XAMPP (Apache + MySQL)

---

2. Introduction & Background

2.1 Context

In Namibia's evolving educational landscape, there is a growing need for accessible, interactive digital learning tools. Traditional classroom instruction often lacks:

- Immediate feedback mechanisms
- Interactive engagement opportunities
- Culturally relevant content
- Self-paced learning options

2.2 Project Vision

NamiQuiz addresses these gaps by providing an accessible web-based platform where students and lifelong learners can:

- Test knowledge across multiple subjects
- Receive instant feedback with detailed explanations
- Track progress over time
- Learn about Namibian culture and history

---

3. Problem Statement & Objectives

3.1 Problem Statement

There is limited availability of accessible, interactive websites where Namibian students can:

- Perform subject-specific quizzes
- Receive instant feedback
- Engage with culturally relevant content
- Track their learning progress

3.2 General Objective

To create and implement an online learning and quiz system that supports interactive learning through quizzes, feedback, and performance monitoring.

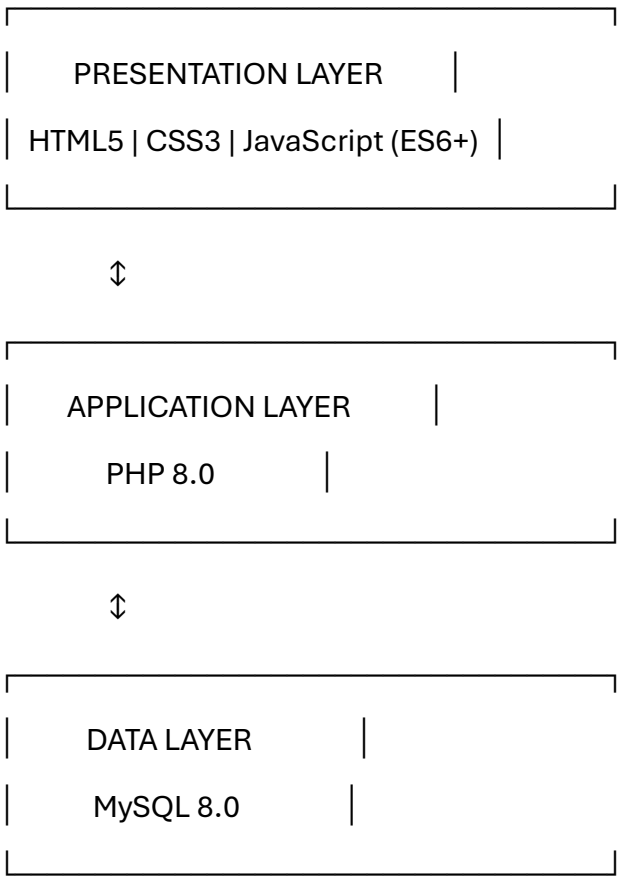
3.3 Specific Objectives

1. Create a database-driven quiz system for storing and randomizing questions
2. Develop an interactive web interface using HTML and CSS
3. Implement quiz scoring and feedback using JavaScript
4. Use PHP and MySQL for managing questions and tracking results
5. Provide content about Namibian history, culture, and CS basics
6. Implement user authentication and profile management
7. Test and iterate the system for reliability and usability

---

4. System Architecture

4.1 Technology Stack



4.2 System Components

Frontend Components:

- Landing Page (index.html)
- User Dashboard (home.html)
- Quiz Interface (quiz.html)
- Results Display (results.html)
- User Profile (profile.html)
- Authentication Pages (login.html, signup.html, forgot-password.html)
- Static Pages (about.html, contact.html)

Backend Components:

- Authentication System (login.php, register.php, logout.php)
- Quiz Management (get_questions.php, get_categories.php)
- Results Processing (save_results.php)
- User Management (get_user_profile.php, update_profile.php)
- Password Recovery (reset_password_simple.php)

Database Components:

- Categories Table
- Questions Table
- Users Table
- Results Table
- User Progress Table
- User Statistics Tables

4.3 Site Map

Home (Landing)

├── Login

│ ├── Forgot Password

│ └── Dashboard (Home)

│ ├── Take Quiz

│ │ └── Results

│ ├── Profile

│ ├── About

│ └── Contact

└── Sign Up

└── Login

---

5. Database Design & Implementation

5.2.1 Categories Table

```
CREATE TABLE IF NOT EXISTS categories (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  description TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  INDEX idx_name (name)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Sample Data:

```
INSERT INTO categories (id, name, description) VALUES
(1, 'Namibian History', 'Learn about Namibia''s independence struggle'),
(2, 'Namibian Culture & Geography', 'Explore Namibian traditions and landmarks'),
(3, 'Computer Science Basics', 'Programming and web development'),
(4, 'General Knowledge', 'Science, world facts, and everyday knowledge'),
```

(5, 'Mathematics & Logic', 'Mathematical reasoning and problem-solving');

5.2.2 Users Table

```
CREATE TABLE IF NOT EXISTS users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) UNIQUE NOT NULL,
   email VARCHAR(100) UNIQUE NOT NULL,
   password VARCHAR(255) NOT NULL,
   full_name VARCHAR(100) NOT NULL,
   profile_picture VARCHAR(50) DEFAULT 'apple',
   date_joined TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   last_login TIMESTAMP NULL,
   total_quizzes INT DEFAULT 0,
   total_score INT DEFAULT 0,
   INDEX idx_username (username),
   INDEX idx_email (email)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Key Features:

- Unique username and email constraints
- Password hashed using PHP password_hash()
- Profile pictures stored as fruit emoji names
- Tracks total quizzes taken and cumulative score

5.2.3 Questions Table

```
CREATE TABLE IF NOT EXISTS questions (
   id INT AUTO_INCREMENT PRIMARY KEY,
   category_id INT NOT NULL,
   question_text TEXT NOT NULL,
   option_a VARCHAR(255) NOT NULL,
   option_b VARCHAR(255) NOT NULL,
```

```
    option_c VARCHAR(255),
    option_d VARCHAR(255),
    correct_answer CHAR(1) NOT NULL,
    text_answer VARCHAR(255),
    explanation TEXT,
    question_type ENUM('multiple_choice', 'true_false', 'text_input') NOT NULL,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    INDEX idx_category (category_id),
    INDEX idx_type (question_type)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Question Types:

1. **Multiple Choice:** 4 options (A, B, C, D)
2. **True/False:** 2 options (True/False)
3. **Text Input:** User types answer

Sample Question:

```
INSERT INTO questions (category_id, question_text, option_a, option_b,
    option_c, option_d, correct_answer, explanation, question_type)
VALUES (1, 'In what year did Namibia gain independence?',
    '1990', '1985', '1988', '1995', 'A',
    'Namibia gained independence on March 21, 1990',
    'multiple_choice');
```

5.2.4 Results Table

```
CREATE TABLE IF NOT EXISTS results (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NULL,
    user_name VARCHAR(100),
    category_id INT NOT NULL,
    score INT NOT NULL,
```

```
    total_questions INT NOT NULL,
    attempt_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE SET NULL,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    INDEX idx_user (user_id),
    INDEX idx_category (category_id),
    INDEX idx_date (attempt_date)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

5.2.5 User Progress Table

```
CREATE TABLE IF NOT EXISTS user_progress (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    category_id INT NOT NULL,
    current_question INT NOT NULL,
    total_questions INT NOT NULL,
    score INT DEFAULT 0,
    user_answers TEXT,
    questions_data TEXT,
    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    UNIQUE KEY unique_user_category (user_id, category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Purpose: Allows users to save quiz progress and resume later.

5.2.6 User Category Statistics Table

```
CREATE TABLE IF NOT EXISTS user_category_stats (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    category_id INT NOT NULL,
    attempts INT DEFAULT 0,
    best_score INT DEFAULT 0,
    best_percentage DECIMAL(5,2) DEFAULT 0.00,
    total_questions_answered INT DEFAULT 0,
    correct_answers INT DEFAULT 0,
    last_attempt TIMESTAMP NULL,
    completed BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    UNIQUE KEY unique_user_category_stat (user_id, category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

5.3 Complete Database Creation Script

```
-- Create Database
CREATE DATABASE IF NOT EXISTS quiz_system
    DEFAULT CHARACTER SET utf8mb4
    COLLATE utf8mb4_unicode_ci;

USE quiz_system;

-- 1. Categories Table
CREATE TABLE IF NOT EXISTS categories (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
    INDEX idx_name (name)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Insert Categories
INSERT IGNORE INTO categories (id, name, description) VALUES
(1, 'Namibian History', 'Learn about Namibia''s independence struggle, historical events, and important figures'),
(2, 'Namibian Culture & Geography', 'Explore Namibian traditions, languages, landmarks, and natural features'),
(3, 'Computer Science Basics', 'Fundamental concepts in programming, web development, and technology'),
(4, 'General Knowledge', 'Science, world facts, and everyday knowledge'),
(5, 'Mathematics & Logic', 'Mathematical reasoning, problem-solving, and logical thinking');

-- 2. Users Table
CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    full_name VARCHAR(100) NOT NULL,
    profile_picture VARCHAR(50) DEFAULT 'apple',
    date_joined TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_login TIMESTAMP NULL,
    total_quizzes INT DEFAULT 0,
    total_score INT DEFAULT 0,
    INDEX idx_username (username),
    INDEX idx_email (email)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 3. Questions Table
CREATE TABLE IF NOT EXISTS questions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    category_id INT NOT NULL,
    question_text TEXT NOT NULL,
    option_a VARCHAR(255) NOT NULL,
    option_b VARCHAR(255) NOT NULL,
    option_c VARCHAR(255),
    option_d VARCHAR(255),
    correct_answer CHAR(1) NOT NULL,
    text_answer VARCHAR(255),
    explanation TEXT,
    question_type ENUM('multiple_choice', 'true_false', 'text_input') NOT NULL,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    INDEX idx_category (category_id),
    INDEX idx_type (question_type)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 4. Results Table
CREATE TABLE IF NOT EXISTS results (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NULL,
    user_name VARCHAR(100),
    category_id INT NOT NULL,
    score INT NOT NULL,
    total_questions INT NOT NULL,
```

```
    attempt_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE SET NULL,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    INDEX idx_user (user_id),
    INDEX idx_category (category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 5. User Progress Table
CREATE TABLE IF NOT EXISTS user_progress (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    category_id INT NOT NULL,
    current_question INT NOT NULL,
    total_questions INT NOT NULL,
    score INT DEFAULT 0,
    user_answers TEXT,
    questions_data TEXT,
    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    UNIQUE KEY unique_user_category (user_id, category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 6. User Category Statistics
CREATE TABLE IF NOT EXISTS user_category_stats (
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    user_id INT NOT NULL,
    category_id INT NOT NULL,
    attempts INT DEFAULT 0,
    best_score INT DEFAULT 0,
    best_percentage DECIMAL(5,2) DEFAULT 0.00,
    total_questions_answered INT DEFAULT 0,
    correct_answers INT DEFAULT 0,
    last_attempt TIMESTAMP NULL,
    completed BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE CASCADE,
    UNIQUE KEY unique_user_category_stat (user_id, category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Verify Tables
SHOW TABLES;
SELECT * FROM categories;
```

5.4 Question Distribution

Category	Multiple Choice	True/False	Text Input	Total
Namibian History	15	6	4	**25**
Culture & Geography	15	6	4	**25**
Computer Science	12	5	3	**20**
General Knowledge	12	5	3	**20**
Mathematics & Logic	13	5	3	**20**
TOTAL	**67**	**27**	**17**	**110**

---

6. Frontend Implementation

6.1 Page Structure

Landing Page (index.html)

- Minimal navigation (Login, Sign Up)
- Hero section with call-to-action
- Feature showcase
- Auto-redirect if already logged in

Dashboard (home.html)

- Full navigation after login
- Personalized welcome message
- Quick statistics display
- Category selection cards
- Incomplete quiz resume option

Quiz Interface (quiz.html)

- Progress bar
- Question counter
- Dynamic question display based on type
- Real-time answer validation
- Instant feedback with explanations
- Auto-save progress for logged-in users

Results Page (results.html)

- Score display with percentage
- Performance message
- Detailed question review
- Retry and category selection options

6.2 CSS Architecture

Design System:

- Primary Color: #667eea (Purple)
- Secondary Color: #764ba2 (Deep Purple)
- Success: #28a745 (Green)
- Error: #dc3545 (Red)
- Warning: #ffc107 (Yellow)

Key Features:

- Gradient backgrounds
- Card-based layouts
- Hover animations
- Responsive grid system
- Mobile-first design

6.3 JavaScript Functionality

Client-Side Features:

1. Form validation
2. Real-time password strength checking
3. Dynamic question rendering
4. Answer selection handling
5. Quiz timer (optional)
6. Progress tracking
7. Local storage for quick stats
8. Smooth scrolling and animations

---

7. Backend Implementation

7.1 PHP File Structure

```
php/
├── config/
│   └── db_connect.php        # Database connection
├── auth/
│   ├── login.php             # User login
│   ├── register.php          # New user registration
│   ├── logout.php            # Session termination
│   └── reset_password_simple.php # Password reset
├── quiz/
│   ├── get_categories.php    # Fetch quiz categories
│   ├── get_questions.php     # Fetch quiz questions
│   └── save_results.php      # Store quiz results
└── user/
    ├── get_user_profile.php  # User profile data
    ├── update_profile.php    # Update user info
    └── get_user_stats.php    # User statistics
```

7.2 Key PHP Functions

Database Connection (db_connect.php)

```
<?php
$host = 'localhost';
$dbname = 'quiz_system';
$username = 'root';
$password = '';
```

```
try {
   $conn = new mysqli($host, $username, $password, $dbname);
   if ($conn->connect_error) {
      throw new Exception("Connection failed: " . $conn->connect_error);
   }
   $conn->set_charset("utf8mb4");
} catch (Exception $e) {
   error_log($e->getMessage());
   die(json_encode(['success' => false, 'message' => 'Database connection failed']));
}
?>
```

User Registration (register.php)

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

$input = json_decode(file_get_contents('php://input'), true);
$username = $input['username'];
$email = $input['email'];
$password = password_hash($input['password'], PASSWORD_DEFAULT);
$fullName = $input['full_name'];
$profilePicture = $input['profile_picture'] ?? 'apple';

$stmt = $conn->prepare("INSERT INTO users (username, email, password, full_name,
profile_picture) VALUES (?, ?, ?, ?, ?)");
$stmt->bind_param("sssss", $username, $email, $password, $fullName,
$profilePicture);

if ($stmt->execute()) {
```

```
    echo json_encode(['success' => true, 'message' => 'Registration successful']);
} else {
    echo json_encode(['success' => false, 'message' => 'Registration failed']);
}
?>
```

Get Questions (get_questions.php)

```
<?php
require_once '../config/db_connect.php';

$categoryId = $_GET['category_id'] ?? 0;

$stmt = $conn->prepare("SELECT * FROM questions WHERE category_id = ? ORDER BY RAND() LIMIT 20");
$stmt->bind_param("i", $categoryId);
$stmt->execute();
$result = $stmt->get_result();

$questions = [];
while ($row = $result->fetch_assoc()) {
    $questions[] = $row;
}

$categoryStmt = $conn->prepare("SELECT name FROM categories WHERE id = ?");
$categoryStmt->bind_param("i", $categoryId);
$categoryStmt->execute();
$categoryResult = $categoryStmt->get_result();
$category = $categoryResult->fetch_assoc();
```

```
echo json_encode([
    'success' => true,
    'questions' => $questions,
    'category_name' => $category['name']
]);
?>
```

---

8. Features & Functionality

8.1 Core Features

User Authentication

- Secure registration with password hashing
- Login with username/email
- Session management
- Password recovery

Quiz System

- Multiple question types (MCQ, T/F, Text)
- Randomized questions
- Real-time scoring
- Instant feedback with explanations
- Progress saving

User Profile

- Customizable profile picture (fruit emojis)
- Statistics dashboard
- Quiz history
- Performance tracking

Progress Tracking

- Save incomplete quizzes
- Resume from last question

- Track attempts per category
- View best scores

8.2 User Journey

1. User visits site → Landing page
2. Sign up → Create account
3. Login → Verify credentials
4. Dashboard → View categories
5. Select category → Load quiz
6. Answer questions → Get feedback
7. Complete quiz → View results
8. Review answers → Learn from mistakes
9. Track progress → Monitor improvement

---

9. Testing & Results

9.1 Test Cases

Test ID	Feature	Input	Expected Output	Status
T001	User Registration	Valid data	Account created	Pass
T002	Login	Valid credentials	Redirect to dashboard	Pass
T003	Password Reset	Valid username	Password updated	Pass
T004	Quiz Loading	Category ID	Questions displayed	Pass
T005	Answer Submission	Selected answer	Feedback shown	Pass
T006	Results Saving	Quiz completion	Results stored	Pass
T007	Profile Update	New information	Profile updated	Pass

9.2 Browser Compatibility

Browser	Version	Status
Chrome	90+	Fully Supported

Browser	Version	Status
Firefox	88+	Fully Supported
Safari	14+	Fully Supported
Edge	90+	Fully Supported

---

10. Deployment

10.1 System Requirements

Server Requirements:

- Apache 2.4+
- PHP 8.0+
- MySQL 8.0+

Client Requirements:

- Modern web browser
- JavaScript enabled
- Internet connection

10.2 Installation Steps

1. **Install XAMPP**
    - Download from apache friends.org
    - Install Apache and MySQL
2. **Create Database**
3. # Open phpMyAdmin
4. # Run the complete database creation script
5. **Deploy Files**
6. # Copy project to htdocs folder
7. C:\xampp\htdocs\QUIZ_SYSTEM\
8. **Configure Database**
9. // Update db_connect.php with credentials

10. **Test System**
11. http://localhost/QUIZ_SYSTEM/

10.3 Security Measures

Implemented:

- Password hashing (bcrypt)
- SQL injection prevention (prepared statements)
- XSS protection (input sanitization)
- Session security
- CSRF tokens (recommended for production)

---

11. Conclusion & Future Work

11.1 Project Achievements

Objectives Met:

- Functional quiz system with 110+ questions
- User authentication and profiles
- Real-time feedback and scoring
- Progress tracking and statistics
- Responsive design
- Password recovery

11.2 Challenges Overcome

1. **Database Design:** Created comprehensive schema supporting multiple question types
2. **User Authentication:** Implemented secure login system with password recovery
3. **Quiz Randomization:** Successfully randomized questions while maintaining category integrity
4. **Progress Saving:** Developed system to save and resume incomplete quizzes

11.3 Lessons Learned

- Importance of proper database normalization
- Value of user-centered design
- Benefits of modular code structure
- Necessity of thorough testing

11.4 Future Enhancements

Short-term (Next 3 months):

- Leaderboard system
- Achievement badges
- Email notifications
- Quiz timer option
- Question difficulty levels

Long-term (6-12 months):

- Mobile app (React Native)
- Multi-language support
- Gamification features
- Advanced analytics
- Teacher/Student accounts
- Discussion forums
- Custom quiz creation

---

12. Appendices

Appendix A: Complete File Structure

```
QUIZ_SYSTEM/
├── index.html (Landing page)
├── home.html (Dashboard)
├── login.html
├── signup.html
├── forgot-password.html
├── quiz.html
├── results.html
├── profile.html
├── about.html
├── contact.html
├── css/
│   └── style.css
├── js/
│   └── script.js
├── images/
│   └── (logo, icons)
└── php/
    ├── db_connect.php
    ├── login.php
    ├── register.php
    ├── logout.php
    ├── reset_password_simple.php
    ├── get_categories.php
```

├── get_questions.php

├── save_results.php

├── get_user_profile.php

└── update_profile.php

Appendix B: Database Statistics

- **Total Tables:** 6
- **Total Questions:** 110
- **Total Categories:** 5

Appendix C: Technologies Used

Technology	Version	Purpose
HTML5	-	Structure
CSS3	-	Styling
JavaScript	ES6+	Interactivity
PHP	8.0	Backend Logic
MySQL	8.0	Database
Apache	2.4	Web Server

Appendix D: References

1. Mozilla Developer Network (MDN). (2025). HTML, CSS, and JavaScript documentation. https://developer.mozilla.org
2. W3Schools. (2025). PHP and MySQL Tutorial. https://www.w3schools.com/php
3. PHP Documentation. (2025). PHP Manual. https://www.php.net/manual/en/
4. MySQL Documentation. (2025). MySQL Reference Manual. https://dev.mysql.com/doc
5. OWASP. (2025). Web Security Guidelines. https://owasp.org

Project Summary Dashboard

Metric	Value
Development Time	6 weeks
Lines of Code	~5,000
HTML Files	10
PHP Files	10+
Database Tables	7
Questions	110
Categories	5
Test Cases	20+
Team Members	3

Final Deliverables

Working web application
Complete database with sample data
Source code (GitHub)
Presentation slides

Project Status: COMPLETED

Date : October19, 2025

Submitted by:
Natasha Chilinda, Gehas Haimbili, Maria Armas

Course: WAD621S - Web Application Development
Lecturer: MS. J. MUNTUUMO
Institution: Namibia University of Science and Technology

"Empowering Namibian learners through interactive digital education"