GAMIFICATION IN EMPLOYEE TRAINING IN CYBERSECURITY USING IOT

ST. TERESA'S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

By

Farhan Zakkeer Hussain- SB22BCA014

III DC BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

Under the guidance of

Ms.Maria Neethu Titus

DEPARTMENT OF CYBER SECURITY AND APPLIED COMPUTING

MARCH 2025

DECLARATION

We, undersigned, hereby declare that the project report, **'GAMIFICATION IN EMPLOYMENT TRAINING IN CYBERSECURITY USING IOT**", submitted for partial fulfillment of the requirements for the award of degree of BCA (Cloud Technology and Information Security Management) at St. Teresa's College (Autonomous), Ernakulam (Affiliated to Mahatma Gandhi University), Kerala, is a bonafide work done by us under the supervision of **Ms. Maria Neethu Titus.** This submission represents our ideas in our own words and where ideas or words of others have not been included. We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Ernakulam Farhan Zakkeer Hussain – SB22BCA014

March 2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

DEPARTMENT OF CYBER SECURITY AND APPLIED COMPUTING

CERTIFICATE

This is to certify that the report entitled "**GAMIFICATION IN EMPLOYEE TRAINING IN CYBERSECURITY USING IOT**", submitted by Farhan Zakkeer Hussain to the Mahatma Gandhi University in partial fulfillment of the requirements for the award of the Degree of BCA (Cloud Computing and Information Security management) is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

ARCHANA MENON P

Head of the Department

21/03/25

Internal Supervisor

Department of Cyber Security and Applied Computing ★ St. Teresa's College (Autonomous) Ernakulam

External Supervisor

ACKNOWLEDGEMENT

First and foremost, we thank God Almighty for his blessings. We take this opportunity to express our gratitude to all those who helped us in completing this project successfully. I wish to express our sincere gratitude to the Directors **Rev. Sr. Tessa CSST** and **Rev. Sr Francis Ann CSST** and the Principal **Dr. Alphonsa Vijiya Joseph** for providing all the facilities.

We express our sincere gratitude to the Head of the Department, **Ms. Archana Menon P**, for the support. We deeply express sincere thanks to our project guide **Ms. Maria Neethu Titus** for her proper guidance and support throughout the project work.

We are indebted to our beloved teachers whose cooperation and suggestion throughout the project which helped us a lot. We thank all our friends and classmates for their support.

We convey our hearty thanks to our parents for the moral support, suggestion and encouragement.

.

ABSTRACT

In today's digital era, cybersecurity awareness is a critical requirement for organizations. Traditional training methods, relying primarily on theoretical instruction, often fail to fully engage employees and lack practical applicability. This project addresses these limitations by developing an innovative gamified training system that integrates Internet of Things (IoT) technology. The system is designed to simulate real-world cybersecurity attacks, such as Trojan Horses, through interactive game levels that allow employees to gain hands-on experience in identifying and mitigating threats. Moreover, IoT-enabled bonus games are incorporated between levels to enhance engagement and provide a tangible learning experience. These bonus games, controlled via Arduino boards and sensors, create a seamless interaction between the digital and physical worlds. By leveraging gamification elements such as leaderboards, badges, and rewards, this approach ensures employees remain motivated while improving skill retention. This system not only engages users but also equips them with critical cybersecurity skills, fostering a proactive approach to organizational security.

TABLE OF CONTENTS

Chapter 1 : INTRODUCTION …… 1

1.1 Role of gamification in Enhancing cybersecurity training …..……..….... 1

1.2 Leveraging Iot technology for Interactive Cybersecurity Simulations……. 2

Chapter 2 : LITERATURE REVIEW……. 3

Chapter 3 : EXISTING SYSTEM AND ITS DRAWBACKS…….. 4

Chapter 4 : PROPOSED SYSTEM AND ITS OBJECTIVES……..... 6

Chapter 5 : SYSTEM REQUIREMENTS……..……8

Chapter 6 : SYSTEM DESIGN AND ARCHITECTURE……..……12

6.1 Key design patterns……15

Chapter 7 : MODULE DESCRIPTION…….16

Chapter 8 : IMPLEMENTATION…….20

Chapter 9 : RESULT AND ANALYSIS……22

REFERENCES

CHAPTER 1
INTRODUCTION

The rapid evolution of cybersecurity threats presents significant challenges for organizations worldwide. As cyberattacks become increasingly sophisticated, companies face a constant battle to protect their sensitive information. While many organizations invest in training programs to mitigate these risks, traditional training methods, such as lectures, theoretical presentations, and quizzes, often fall short. These approaches lack interactivity and fail to simulate the complexities of real-world cyberattacks, leaving employees underprepared to handle actual security threats.

To address this issue, gamification has emerged as an effective solution. Gamification involves the integration of game mechanics—such as levels, points, rewards, and challenges—into non-gaming contexts. By introducing these elements into cybersecurity training, gamification boosts engagement and motivates employees to actively participate in the learning process. Employees are more likely to retain information when they are incentivized to reach higher levels or earn rewards for completing tasks, which makes the training more engaging and enjoyable. Additionally, the competitive and goal-oriented nature of gamification encourages employees to improve their skills continuously.

Moreover, combining gamification with IoT (Internet of Things) technology can take the training experience a step further. IoT devices allow for interactive, hands-on learning by incorporating physical activities alongside the virtual elements of the game. For example, employees could interact with connected devices, such as smart locks or security cameras, to identify and respond to potential security vulnerabilities. This not only deepens their understanding of cybersecurity concepts but also provides them with the practical experience needed to tackle real-world threats.

Incorporating both gamification and IoT into cybersecurity training helps organizations build a more effective, engaging, and immersive learning environment. Employees are more likely to develop the necessary skills and awareness to respond to emerging threats and contribute to the overall security of their organizations. By moving beyond traditional methods and embracing these innovative approaches, companies can better prepare their teams for the ever-evolving landscape of cybersecurity challenges.

1.1. The Role of Gamification in Enhancing Cybersecurity Training

Gamification, by incorporating game mechanics such as points, levels, and rewards, transforms traditional cybersecurity training into an engaging and interactive experience. This subtopic can explore how gamification fosters active participation, improves retention of cybersecurity knowledge, and motivates employees to stay engaged with ongoing training. It could also address the competitive and goal-oriented elements of gamification, which encourage employees to continuously improve their skills.

1.2. Leveraging IoT Technology for Interactive Cybersecurity Simulations

The integration of IoT technology into cybersecurity training creates an immersive, hands-on learning environment. This subtopic could discuss how IoT-enabled devices allow employees to interact with real-world security systems, such as smart devices or network security equipment, as part of their training. By simulating real-world scenarios, IoT-based training helps employees develop practical problem-solving skills to better prepare them for potential cyber threats.

CHAPTER 2
LITERATURE REVIEW

 Lasha Abuladze (2021) – Focuses on how gamification can enhance employee training and improve soft skills development, making learning more engaging and effective.

 Armstrong and Landers (2018) – Explore how gamification, with elements like rewards and badges, motivates employees and reinforces learning outcomes, enhancing training programs.

 Mollick and Rothbard (2013) – Introduce the idea of 'mandatory fun,' showing how gamification transforms routine tasks into enjoyable and meaningful activities that increase engagement.

 Silic et al. (2020) – Investigate the influence of gamified HR systems on job satisfaction and employee engagement, highlighting how game elements can improve workplace dynamics.

 Blohm and Leimeister (2013) – Offer a framework for implementing gamification, stressing the need to tailor game mechanics to align with the company's goals and objectives.

 Oprescu et al. (2014) – Propose ten principles for effectively gamifying workplace processes, ensuring that gamification strategies are well-executed and yield desired results.

 Larson (2020) – Reviews the use of serious games in corporate training, discussing their role in fostering skill development and knowledge retention in a more interactive way.

 Ahmed and Sutton (2017) – Examine the integration of gamification with advanced technologies like IoT and augmented reality, exploring how these technologies can enhance employee training experiences.

 Pereira et al. (2018) – Focus on the application of gamification in creating interactive work instructions, showing how game-like elements can enhance clarity and engagement in learning materials.

CHAPTER 3
EXISTING SYSTEM AND ITS DRAWBACKS

The current methods employed in cybersecurity training predominantly rely on theoretical instruction, presentations, and quizzes. While these approaches are informative, they often fall short in equipping employees with the practical skills needed to handle real-world threats effectively. The focus on delivering information rather than fostering practical application creates gaps in the training process, making it difficult for employees to connect theoretical concepts with real-life scenarios. As a result, knowledge retention can be limited, and employees may struggle to recall or apply security protocols when faced with live threats.

Additionally, traditional cybersecurity training programs are typically standardized and fail to accommodate the diverse learning needs or styles of employees. Each individual learns differently, but traditional programs tend to use a one-size-fits-all approach, which can leave some employees feeling disengaged or unchallenged. This lack of personalization can make it harder for employees to retain important security practices and reduces the overall effectiveness of the training.

Furthermore, the absence of immediate feedback within conventional training formats means that employees are not able to quickly identify and correct any mistakes they make during the learning process. In cybersecurity, where decisions need to be made swiftly and accurately, the inability to receive feedback in real-time can hinder the development of critical problem-solving skills. Employees may not fully grasp the consequences of their actions in a simulated environment, which diminishes their readiness to respond effectively to real threats in a high-pressure situation.

In light of these limitations, there is an increasing demand for more interactive, engaging, and practical approaches to cybersecurity training. By incorporating hands-on activities, simulations, and real-world threat scenarios, organizations can better prepare employees to respond to the ever-evolving landscape of cyber threats. Additionally, integrating personalized learning paths that cater to individual needs and providing real-time feedback during training can enhance both knowledge retention and overall effectiveness.

Moving towards more dynamic training methods will not only improve employee engagement but also ensure that they are better equipped to handle cybersecurity challenges when they arise.

2.1.DRAWBACKS

1. Lack of Real-World Simulation

Traditional training often fails to simulate actual cyberattacks, leaving employees unprepared for the unpredictable nature of real-world threats.

2. Limited Engagement

Lectures and passive learning formats can lead to disengagement, making it harder for employees to retain information and stay motivated throughout the training.

3. Outdated Content

Cybersecurity threats evolve rapidly, and traditional training may fail to keep up with emerging attack techniques, leaving employees ill-prepared for new risks.

4. One-Size-Fits-All Approach

Standardized training doesn't cater to the different learning styles or knowledge levels of employees, which can result in ineffective learning for some individuals.

5. Lack of Hands-On Practice

Traditional methods focus on theory but lack practical exercises that allow employees to develop and apply real-world cybersecurity skills.

\

CHAPTER 4
PROPOSED SYSTEM AND ITS OBJECTIVES

To address the shortcomings of existing training methods, this project introduces a gamified, web-based cybersecurity training platform integrated with IoT-enabled components for enhanced interactivity. This system seeks to transform traditional cybersecurity training into a dynamic, immersive, and hands-on learning experience. By combining gamification and IoT, the platform offers a multifaceted approach that goes beyond theoretical instruction and delivers practical, real-world skills necessary for responding to ever-evolving cyber threats.

The gamified elements of the platform, such as leaderboards, badges, and progress tracking, are designed to motivate employees and foster a sense of achievement as they progress through various training modules. The use of rewards and recognition systems encourages continuous learning, making the training experience more engaging and enjoyable. By incorporating elements of friendly competition, employees are encouraged to stay committed to their learning goals and are more likely to retain the knowledge they acquire.

Simulating real-world attack scenarios within the training platform allows employees to experience first-hand the challenges of dealing with security breaches, data theft, or system vulnerabilities. These simulations provide a safe environment for employees to practice their response to cybersecurity threats without the risk of causing actual harm to the organization’s systems. In this controlled setting, trainees can develop their problem-solving skills, learn how to detect potential threats, and understand the appropriate steps to mitigate them. This hands-on experience is crucial for bridging the gap between theoretical knowledge and practical application

.

3.1.OBJECTIVES

1. **Enhance Employee Engagement and Motivation**
   Utilize gamification techniques such as leaderboards, badges, and progress tracking to increase employee participation, making cybersecurity training more interactive, rewarding, and enjoyable.
2. **Improve Practical Skills through Real-World Simulations**
   Provide employees with hands-on experience by simulating real-world cybersecurity attack scenarios, allowing them to practice identifying and responding to security threats in a controlled, risk-free environment.
3. **Reinforce Key Concepts with IoT-Enabled Interactive Activities**
   Integrate IoT technology to offer tangible, interactive exercises that help employees understand the security risks posed by connected devices and how to mitigate these risks effectively.

CHAPTER 5
SYSTEM REQUIREMENTS

Hardware Requirements:

● Processor: Any modern CPU (Intel Core i3/AMD Ryzen 3 or better)

● RAM: Minimum 4GB

● Storage: 1GB free space

● Internet Connection: Broadband connection (minimum 1Mbps)

- pressure sensor
- Arduino microcontroller
- wirelessly connected or linked via USB/Bluetooth

Software Requirements:

● Operating System: Cross-platform (Windows/Linux/macOS)

● Python 3.8 or higher

● Modern web browser (Chrome, Firefox, Safari, Edge)

● SQLite database

Package Dependencies:

1. Flask (v2.0.1)

a. Core web framework

b. Implementation Files:

i. app.py: Main application file containing all route handlers and core logic

ii. templates/*.html: All template files using Jinja2 templating

c. Implementation Details:

i. Route handling for all endpoints (@app.route decorators)

ii Template rendering using render_template()

iii. Request handling using request object

iv. Session management using session object

v. Flash messages using flash()

2. Flask-SQLAlchemy (v2.5.1)

a. Database ORM

b. Implementation Files:

i. app.py: Database models and queries

ii. init_db.py: Database initialization and schema setup

c. Implementation Details:

i. User model with authentication fields

ii. Level model for game progression

iii. Relationship definitions between models

iv. Query operations for user progress tracking

3. SQLAlchemy (v1.4.46)

a. SQL Toolkit

b. Implementation Files:

i. . app.py: Model definitions and database operations

ii init_db.py: Schema creation and initial data setup

c. Implementation Details:

i. Database model definitions using db.Model

ii. Column type definitions and constraints

iii. Foreign key relationships

iv. Complex queries for level progression

4. Flask-Login (v0.5.0)

a. User session management

b. Implementation Files:

i. . app.py: User authentication and session handling

ii. templates/register.html: Registration form

iii. templates/login.html: Login form

iv. templates/profile.html: Profile update form

c.Implementation Details:

i. User authentication with @login_required decorator

ii. User session management with login_user() and logout_user()

iii. Current user access with current_user

5. Flask-WTF (v1.2.1)

a. Form handling and validation

b. Implementation Files:

i. app.py: Form definitions and validation

ii. templates/register.html: Registration form

iii. templates/login.html: Login form

iv. templates/profile.html: Profile update form

c. Implementation Details:

i. CSRF protection for all forms

ii. Form validation rules

iii.Custom validators for security requirements

6. Werkzeug (v2.0.1)

a. WSGI utility library

b. Implementation Files:

i. app.py: Password hashing and file handling

c. Implementation Details:

i. . Password hashing with generate_password_hash()

ii Password verification with check_password_hash()

iii. Secure filename handling with secure_filename()

7. email-validator (v2.1.0.post1)

a. Email validation

b. Implementation Files:

i. . app.py: Registration and profile update routes

ii templates/register.html: Registration form

c. Implementation Details:

i. Email format validation

ii. Domain validation

iii. MX record checking

8. python-dotenv (v1.0.0)

a. Environment variable management

b. Implementation Files:

i. .env: Environment variables

ii. app.py: Configuration loading

c. Implementation Details:

i. Secret key management

ii. Database URL configuration

iii. Environment-specific settings

9. pyotp (v2.6.0)

a. Two-factor authentication

b. Implementation Files:

i. app.py: 2FA routes and verification

ii. templates/verify_2fa_login.html: 2FA verification page

c. Implementation Details:

i. TOTP secret generation

ii. Token verification

iii. Backup code management

10. Pillow (v11.1.0)

a. Image processing

b. Implementation Files:

i. app.py: Profile picture upload routes

ii. templates/profile.html: Profile picture form

c. Implementation Details:

i. Image resizing and optimization

ii. Format validation

iii. Secure image storage

11. qrcode (v8.0)

a. QR code generation

b. Implementation Files:

i. app.py: 2FA setup routes

ii. templates/profile.html: 2FA setup section

c. Implementation Details:

i. QR code generation for 2FA setup

ii. Base64 encoding for display

iii. TOTP URI generation

CHAPTER 6
SYSTEM DESIGN AND ARCHITECTURE

Architecture Components:

1. Models Layer

The Models Layer serves as the foundation of the application by defining the core data structures and their relationships. It uses SQLAlchemy ORM to interact with an SQLite database, ensuring seamless data management.

a. User Model

- Stores user-related data such as username, email, password hash, and profile information.
- Handles user authentication, registration, and login mechanisms.
- Implements role-based access control (RBAC) for different user roles such as players and administrators.
- Uses password hashing techniques such as bcrypt for secure storage of credentials.
- Provides methods for updating user profiles and managing user preferences.

b. Level Model

- Manages game levels, including metadata such as level number, difficulty, and requirements.
- Tracks user progression by maintaining records of completed levels and achievements.
- Stores level-specific configurations, such as time limits, scoring parameters, and available resources.
- Supports dynamic difficulty adjustment based on player performance.
- Integrates with the scoring system to update and maintain leaderboards.

c. Database Schema (SQLite with SQLAlchemy ORM)

- Uses SQLite as the relational database management system for lightweight and efficient storage.
- Implements SQLAlchemy ORM to facilitate database operations using Python objects.

- Defines relationships between tables to enforce referential integrity.
- Supports migrations to handle schema changes without data loss.
- Optimizes query performance through indexing and caching mechanisms.

2. View Layer

The View Layer is responsible for rendering the user interface and ensuring a seamless interaction between the user and the application.

a. Templates (Jinja2-based HTML templates)

- Uses Jinja2 templating engine to dynamically generate HTML pages.
- Provides reusable components such as headers, footers, and navigation bars.
- Supports conditional rendering and looping constructs for efficient content generation.
- Implements CSRF protection in forms to enhance security.
- Ensures responsiveness and accessibility through well-structured HTML.

b. Static Assets (CSS, JavaScript, and Media Files)

- Includes CSS files for styling the user interface and ensuring a consistent visual design.
- Uses JavaScript for client-side interactivity, including AJAX requests and dynamic content updates.
- Stores media assets such as images, icons, and audio files required for the game environment.
- Implements asset versioning to prevent caching issues and ensure smooth updates.
- Organizes assets in a structured manner for easy management and maintainability.

c. Frontend Components (Responsive UI elements)

- Designs responsive UI elements that adapt to different screen sizes and resolutions.
- Implements interactive elements such as buttons, sliders, and modals for enhanced user experience.
- Uses frameworks like Bootstrap or Tailwind CSS to streamline development.
- Ensures accessibility compliance to support users with disabilities.
- Optimizes front-end performance by minimizing render-blocking resources.

3. Controller Layer

The Controller Layer acts as the intermediary between the Model and View Layers, processing user requests and managing application logic.

a. Route Handlers (Process requests and manage application flow)

- Defines endpoints for handling user requests, such as login, signup, and game progression.
- Uses Flask or a similar framework to map URLs to corresponding controller functions.
- Implements middleware for request validation and error handling.
- Supports API routes for AJAX-based interactions and mobile integrations.
- Handles session management and user state tracking.

b. Authentication Controllers (Handle user authentication)

- Manages user login, logout, and session handling mechanisms.
- Implements multi-factor authentication (MFA) for enhanced security.
- Supports social login integrations such as Google or Facebook authentication.
- Handles password reset and account recovery functionalities.
- Logs authentication attempts and provides audit trails for security monitoring.

c. Game Logic Controllers (Manage level progression and scoring)

- Implements game mechanics, including rules for progression and scoring.
- Tracks player performance and updates level status accordingly.
- Manages power-ups, rewards, and penalties based on gameplay actions.
- Integrates with external APIs for leaderboard and multiplayer functionalities.
- Ensures fair play by implementing anti-cheating mechanisms.

6.1.Key Design Patterns:

1. Authentication Flow:

User Input → Password Hashing → Database Validation → Session Management → 2FA Verification

2. Level Progression System:

Level Selection → Challenge Presentation → User Input → Validation → Points Award → Progress Update

3. Security Layer:

Request → CSRF Protection → Authentication Check → 2FA Verification → Authorization → Response

CHAPTER 7

MODULE DESCRIPTION

1. Authentication Module

- **Implementation Files**:
  - app.py: Routes and logic for authentication
  - templates/login.html: Login interface
  - templates/register.html: Registration interface
  - templates/verify_2fa_login.html: 2FA verification
- **Key Functions**:
  - login(): Handles user authentication
  - register(): New user registration
  - logout(): Session termination
  - verify_2fa_login(): 2FA verification
- **Database Tables**:
  - User model with authentication fields
- **Security Features**:
  - Password hashing
  - CSRF protection
  - Rate limiting
  - Session management

2. Two-Factor Authentication Module

- **Implementation Files**:
  - app.py: 2FA setup and verification routes
  - templates/profile.html: 2FA management interface
  - templates/verify_2fa_login.html: 2FA verification
- **Key Functions**:
  - setup_2fa(): Initial 2FA configuration
  - verify_2fa(): Token verification
  - generate_backup_codes(): Recovery code generation
  - disable_2fa(): 2FA deactivation
- **Database Fields**:
  - two_factor_enabled
  - two_factor_secret
  - two_factor_backup_codes

- **Security Features**:

  - TOTP implementation
  - Rate limiting
  - Backup code system

3. User Profile Module

- **Implementation Files**:
  - `app.py`: Profile management routes
  - `templates/profile.html`: Profile interface
  - `static/uploads/`: Profile picture storage
- **Key Functions**:
  - `update_profile()`: Profile information updates
  - `upload_profile_picture()`: Image handling
  - `change_password()`: Password management
  - `update_security_settings()`: Security preferences
- **Database Fields**:
  - Profile information
  - Security settings
  - Profile picture path
- **Features**:
  - Image processing
  - Form validation
  - Security settings management

4. Game Level Module

- **Implementation Files**:
  - `app.py`: Level management routes
  - `templates/levels/`: Level templates
  - `templates/levels/user/`: User mode levels
  - `templates/levels/developer/`: Developer mode levels
  - `templates/levels/base_level.html`: Base level template
- **Key Functions**:
  - `level()`: Level rendering and progression
  - `submit_level()`: Answer validation
  - `update_user_progress()`: Progress tracking
  - `calculate_points()`: Scoring system
- **Database Tables**:
  - Level progression tracking
  - User points system
  - Completion status

- **Features**:

- Progressive difficulty
- Score tracking
- Achievement system

5. User Interface Module

- **Implementation Files**:
  - templates/base.html: Base template
  - templates/dashboard.html: Main interface
  - templates/levels.html: Level selection
  - static/css/: Stylesheets
  - static/js/: JavaScript files
- **Key Components**:
  - Responsive navigation
  - Progress visualization
  - Interactive elements
  - Error handling
- **Features**:
  - Mobile-responsive design
  - Dynamic content loading
  - Real-time feedback
  - User Experience Optimization

6.BONUS GAME

Integration of Pressure Sensor Shoe for Gameplay

The bonus game in the Cybersecurity Training Platform integrates a pressure sensor shoe connected to an Arduino microcontroller. This setup enables real-world physical movement to control in-game character actions, enhancing immersion and engagement.

1. Hardware Setup

- A pressure sensor is embedded in the shoe to detect foot movements (jumps, runs, and steps).
- The Arduino microcontroller acts as a keyboard emulator, sending input signals to the game.
- The setup is wirelessly connected or linked via USB/Bluetooth to transmit real-time data.

2. Functionality in the Bonus Game

- When the employee jumps, the in-game character also jumps.
- Running in place accelerates movement in the game.
- The sensor detects pressure changes, converting them into game commands.
- This feature enhances engagement, physical activity, and gamification in training.

3. Implementation in the System

- The Arduino reads pressure values from the shoe sensor.
- If pressure exceeds a certain threshold (jump detection), a keypress event (e.g., Spacebar for jump) is sent to the game.
- Continuous pressure changes (running detection) are mapped to movement keys (e.g., Left/Right Arrows).
- The system ensures real-time responsiveness, making the interaction seamless and immersive.

We acknowledge “JUMP-GUYS” from CRAZY GAMES for providing inspiration and content used in this project. The game is available at https://www.crazygames.com/game/jump-guys

CHAPTER 8
IMPLEMENTATION

Core Authentication Implementation:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        user =
User.query.filter_by(username=request.form['username']).first()
        if user and check_password_hash(user.password_hash,
request.form['password']):
            if user.two_factor_enabled:
                session['pending_user_id'] = user.id
                return redirect(url_for('verify_2fa_login'))
            login_user(user)
            return redirect(url_for('dashboard'))
    return render_template('login.html')
```

Two-Factor Authentication Implementation:

```
@app.route('/verify_2fa', methods=['POST'])
@login_required
def verify_2fa():
    totp = pyotp.TOTP(current_user.two_factor_secret)
    if totp.verify(request.form['code']):
        session['2fa_verified'] = True
        return jsonify({'success': True})
    return jsonify({'success': False}), 401
```

Level Management Implementation:

```
@app.route('/level/<int:level_id>')
@login_required
def level(level_id):
    level = Level.query.get_or_404(level_id)
    completed_levels = current_user.completed_levels.split(',') if
current_user.completed_levels else []
    return render_template('levels/base_level.html',
                           level=level,
                           completed=str(level_id) in completed_levels)
```

User Progress Tracking:

```
def update_user_progress(user_id, level_id, points):
    user = User.query.get(user_id)
    if user:
        completed_levels = user.completed_levels.split(',') if
user.completed_levels else []
        if str(level_id) not in completed_levels:
            completed_levels.append(str(level_id))
            user.completed_levels = ','.join(completed_levels)
            user.points += points
            db.session.commit()
```

CHAPTER 9
RESULT AND ANALYSIS

The Cybersecurity Training Game Platform has successfully achieved its primary objectives by delivering an interactive, engaging, and security-focused learning experience. The platform effectively combines game mechanics, security implementations, and educational elements to create a robust and user-friendly training environment.

Through an in-depth evaluation, we analyze the platform's effectiveness across four key areas: enhanced learning experience, security implementation, user engagement, and learning outcomes.

1. Enhanced Learning Experience

The platform is designed to provide an engaging and hands-on learning experience for users of all skill levels. The following factors contribute to an interactive and effective cybersecurity training environment:

a. Interactive Gameplay Mechanics

The training incorporates gamification elements to maintain user interest.

Users progress through real-world-inspired security challenges that require critical thinking and problem-solving skills.

The game offers multiple-choice, hands-on coding challenges, and live security attack simulations to cater to diverse learning preferences.

b. Real-World Scenario Simulations

The game presents practical cybersecurity threats such as phishing attacks, SQL injections, password cracking, and malware analysis.

Users are required to identify, analyze, and mitigate security threats in simulated real-world cybersecurity environments.

Provides incident response simulations where users must quickly react to security breaches.

c. Progressive Difficulty Levels

The game follows a structured progression system, where challenges increase in complexity.

Users start with basic cybersecurity concepts and gradually move to advanced hacking and security exploitation techniques.

Unlocking higher levels requires completing the previous challenges, ensuring a structured learning path.

d. Immediate Feedback System

Users receive real-time feedback on their answers and solutions, helping them understand mistakes instantly.

Provides detailed explanations for incorrect answers, promoting self-learning and improvement.

Displays hints and guides for complex problems to enhance user experience and learning retention.

2. Security Implementation

Security is a core focus of the platform, ensuring that both user data and platform integrity remain protected. The following security mechanisms have been effectively implemented:

a. Robust Authentication System

The platform employs a secure authentication mechanism using password hashing (bcrypt) and session tokens.

Ensures secure login and registration processes to prevent unauthorized access.

Implements account lockout mechanisms to protect against brute-force attacks.

b. Two-Factor Authentication (2FA)

Provides an additional layer of security by requiring OTP-based verification before login.

Users can enable TOTP-based authentication (Google Authenticator, Authy).

Generates backup codes for account recovery in case of lost authentication devices.

c. Secure Session Management

Uses secure cookies with HTTPOnly and Secure flags to prevent session hijacking.

Implements automatic session expiration after a period of inactivity.

Prevents cross-site session tampering (CSRF attacks) with token-based security measures.

d. Protected File Uploads

Ensures strict validation of uploaded files to prevent malicious file injections.

Uses server-side validation to block potentially harmful files and scripts.

Implements automatic file scanning to detect and remove security threats.

3. User Engagement

To maintain high levels of user engagement, the platform integrates gamification techniques that encourage users to actively participate and progress through cybersecurity challenges.

a. Point-Based Reward System

Users earn points based on the difficulty and accuracy of their completed challenges.

Points can be used to unlock new levels, hints, and bonus challenges.

Encourages healthy competition by allowing users to compare scores on a leaderboard.

b. Achievement Tracking

The platform includes an achievement system where users can unlock badges and trophies for completing challenges.

Tracks personal milestones, such as completing a certain number of challenges or achieving high scores.

Encourages continuous participation by rewarding users for maintaining a learning streak.

c. Profile Customization

Users can personalize their profiles by setting custom avatars, themes, and preferences.

Allows users to track their progress, achievements, and learning history.

Provides an interactive dashboard for monitoring performance over time.

d. Progress Visualization

Users can track their current progress and skill improvement through visual graphs and performance reports.

Implements progress bars and completion status for each level.

Helps users identify areas of weakness and strengths to focus on.

4. Learning Outcomes

The Cybersecurity Training Game Platform successfully delivers practical knowledge and hands-on experience to users, preparing them for real-world cybersecurity challenges. The learning outcomes include:

a. Practical Cybersecurity Knowledge

Users gain practical skills in network security, ethical hacking, and cryptography.

Provides real-time attack simulations to train users in defensive security strategies.

Helps users understand vulnerabilities and mitigation techniques.

b. Hands-On Experience

The platform allows users to apply theoretical knowledge in an interactive environment.

Users can experiment with security exploits, analyze logs, and implement security defenses.

Encourages hands-on practice instead of relying solely on theoretical learning.

c. Best Practices Understanding

The game emphasizes secure coding practices, strong authentication mechanisms, and data protection strategies.

Users learn how to identify, prevent, and respond to cybersecurity threats.

Reinforces ethical hacking principles and responsible security practices.

d. Security Awareness

Helps users recognize common cybersecurity threats such as phishing, malware, and social engineering attacks.

Trains users to develop a security-first mindset when handling sensitive data.

Promotes awareness of emerging cybersecurity trends and best practices.

REFERENCES

1. Cybersecurity and Gamification Research

Gamification in Cybersecurity Training - https://www.sciencedirect.com/science/article/pii/S1877050919317253

The Effectiveness of Gamification in Cybersecurity Awareness - https://ieeexplore.ieee.org/document/8876629

Gamified Learning for Cybersecurity Education - https://www.tandfonline.com/doi/full/10.1080/10447318.2020.1744577

2. Cybersecurity Training Platforms

TryHackMe (Gamified Cybersecurity Learning) - https://www.tryhackme.com

Hack The Box (Practical Cybersecurity Challenges) - https://www.hackthebox.com

Cybrary (Cybersecurity Learning Platform) - https://www.cybrary.it

3. Authentication and Security Best Practices

OWASP Authentication Cheat Sheet - https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

NIST Cybersecurity Framework - https://www.nist.gov/cyberframework

Google Authentication Best Practices - https://developers.google.com/identity

4. Gamification in Learning

Gamification of Learning and Training - https://elearningindustry.com/benefits-gamification-corporate-training

The Psychology of Gamification in Learning - https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01547/full