

A Project Report On

SmartlyGoPHP

Submitted in partial fulfillment of the requirement for the
award of the degree

Bachelor of Computer Application
BCA

Academic Year 2025 - 26

Jenish Gohel
92300527156

Ravi Sav
92300527257

Internal Guide
Prof. Vipulkumar Bambhaniya



Marwadi
University
Marwadi Chandarana Group





Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled

SmartlyGoPHP

*submitted in partial fulfillment of the requirement for
the award of the degree of
Bachelor of Computer Application*

BCA

of the

Marwadi University

is a result of the bonafide work carried out by

Jenish Gohel (92300527156)

Ravi Sav (92300527257)

during the academic year 2025-26

Prof. Vipulkumar Bambhaniya

Faculty Guide

Dr. Sunil Bajeja

HOD

Dr. Rijwan Khan

Dean

DECLARATION

We hereby declare that this project work entitled **SmartlyGoPHP** is a record done by us.

I also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place : Marwadi University , Rajkot

Date : 25th January , 2025

Jenish Gohel (92300527156)

Signature : Jenish Gohel

Ravi Sav (92300527257)

Signature : Ravi Sav

CONTENTS

Chapters	Particulars	Page No.
1	SYNOPSIS	7
2	PREAMBLE	8
2.1	General Introduction	8
2.2	Module description	9
3	REVIEW OF LITERATURE	11
4	TECHNICAL DESCRIPTION	12
	Hardware Requirement	12
	Software Requirement	13
5	SYSTEM DESIGN AND DEVELOPMENT	
5.1	• Flowchart	14
5.2	• Data Flow Diagram,	16
5.3	• Sequential Diagram	17
5.4	• Use Case Diagram	18
5.5	• Class Diagram	18
5.6	• Database Design	19
5.7	• Screen Design & Coding	24
6	CONCLUSION	45
7	LEARNING DURING SIP	46
8	BIBLIOGRAPHY	
8.1	Online References	
8.2	Offline References	47

TABLE INDEX

Table No.	Title	Page No.
Table 4.1	Hardware Configuration	12
Table 4.2	Software Configuration	13

FIGURE INDEX

Figure No.	Title	Page No.
Figure 5.1	Admin Flow Chart	14
Figure 5.1.2	User Flow Chart	15
Figure 5.2	DFD Level 0	16
Figure 5.2.1	DFD Level 1 Admin	16
Figure 5.2.2	DFD Level 1 User	17
Figure 5.3	Sequential Diagram of Authentication	17
Figure 5.4	Use Case Diagram	18
Figure 5.5	Class Diagram	18
Figure 5.7.1	Admin Login Page	24
Figure 5.7.3	Admin Dashboard Page	26
Figure 5.7.4	Manage Users Page	26
Figure 5.7.5	Admin Edit/Delete User Options	27
Figure 5.7.6	Edit User Details Page	27
Figure 5.7.7	Manage Feedback Page	28
Figure 5.7.8	Manage Contact Page	28
Figure 5.7.9	Student Login Page	29
Figure 5.7.10	Student Registration Page	29
Figure 5.7.11	User Dashboard Page 1	30
Figure 5.7.12	User Dashboard Page 2	30
Figure 5.7.13	User Dashboard Page 3	31
Figure 5.7.14	Tools Page 1	31
Figure 5.7.15	Tools Page 2	32
Figure 5.7.16	Tools Page 3	32
Figure 5.7.17	Fill in the Blanks Page 1	33
Figure 5.7.18	Fill in the Blanks Page 2	33
Figure 5.7.19	Quiz Challenge Page 1	34
Figure 5.7.20	Quiz Challenge Page 2	34
Figure 5.7.21	Quiz Challenge Page 3	35
Figure 5.7.22	True or False Page 1	35
Figure 5.7.23	True or False Page 2	36
Figure 5.7.24	True or False Page 3	36
Figure 5.7.25	Find the Error Page	37

Figure 5.7.26	Fix the Code Page 1	37
Figure 5.7.27	Fix the Code Page 2	38
Figure 5.7.28	PHP Flashcards Page	38
Figure 5.7.29	Predict the Output Page	39
Figure 5.7.30	PHP Treasure Hunt Page	39
Figure 5.7.31	Modules Page 1	40
Figure 5.7.32	Modules Page 2	40
Figure 5.7.33	Modules Page 3	41
Figure 5.7.34	Feedback Page 1	41
Figure 5.7.35	Feedback Page 2	42
Figure 5.7.36	Contact Page 1	42
Figure 5.7.37	Contact Page 2	43
Figure 5.7.38	User Post Logout Page	43
Figure 5.7.39	User without Login Page	44

1. SYNOPSIS

- **Project Title:** SmartlyGoPHP
- **Project Type:** Web-Based PHP Learning Platform
- **Technologies Used:** Core PHP, MySQL, HTML, CSS, JavaScript
- **Overview:**

SmartlyGoPHP is an interactive learning platform developed to help students and beginners learn Core PHP concepts in a simple, practical, and engaging way. The system combines learning content with interactive games and tools to improve understanding and retention.
- **System Structure:**

The project is divided into two main parts:

 1. Admin Side: Used to manage users, monitor feedback and contact messages, and view overall platform statistics through a dashboard.
 2. User Side: Allows learners to register, log in, access PHP learning modules, and use various learning and practice tools.
- **Learning & Gamification:**

Users can learn PHP concepts through modules and test their knowledge using quizzes, fill-in-the-blanks, and true/false games that award points and increase user levels. Additional practice tools like code fixing, flashcards, and output prediction help strengthen programming logic without affecting scores.
- **Data Management:**

The MySQL database securely stores user information, game data, progress details, scores, accuracy, and learning streaks. Guest users can explore the platform without logging in, but their progress is not saved.
- **Objective:**

The main objective of SmartlyGoPHP is to make learning PHP more interactive, practical, and enjoyable while providing structured progress tracking for registered users.

2.PREAMBLE

2.1 GENERAL INTRODUCTION

Education and learning methodologies have undergone a major transformation with the growth of digital technologies. While traditional learning methods such as books, classroom lectures, and written notes are still important, they often lack interactivity, self-assessment, and continuous engagement. In the field of programming education, learners require hands-on practice, instant feedback, and logical problem-solving exposure to truly understand concepts.

With the rapid advancement of web technologies, interactive learning platforms have become an effective solution to bridge the gap between theory and practice. Programming languages like PHP demand not only conceptual understanding but also continuous practice through coding, debugging, and problem-solving exercises.

Relevance of the Project – “SmartlyGoPHP”

The SmartlyGoPHP project addresses these challenges by providing a centralized, interactive, and gamified learning platform focused on Core PHP concepts. Instead of depending on multiple websites or resources for learning, quizzes, and practice tools, learners can access everything through a single web-based system.

The core design philosophy of SmartlyGoPHP includes:

1. Simplicity:

The platform is designed to be user-friendly and easy to understand, making it suitable for beginners as well as intermediate learners.

2. Interactivity:

Learning is enhanced through quizzes, fill-in-the-blanks, true/false questions, and coding-based challenges that keep users actively engaged.

3. Gamification:

A score, level, and streak-based system motivates users to practice consistently and improve their performance.

4. Practical Learning:

Tools such as finding errors, fixing code, predicting output, and flashcards help users strengthen their logical and coding skills.

Broader Impact :

SmartlyGoPHP can be effectively used by college students, beginners in PHP, and self-learners who want to strengthen their Core PHP fundamentals. Educational institutions and training centers can adopt this platform as a supplementary learning tool. In the future, the system can be extended with advanced PHP topics, analytics, leaderboards, and personalized learning paths.

2.2 MODULE DESCRIPTION

The SmartlyGoPHP system is divided into two major modules:

- 1. Admin Module**
- 2. User Module**

1. Admin Module :

The Admin Module acts as the control and monitoring panel of the SmartlyGoPHP application. It allows administrators to manage users, feedback, and contact queries efficiently while keeping track of overall system activity.

Sub-Modules & Functionalities :

1. Authentication

- Admin Login:**
Administrators must log in using valid credentials to access the admin dashboard securely.

2. Dashboard

- Displays total registered users
- Shows total feedback received
- Shows total contact messages

3. User Management

- View complete list of registered users
- Edit user details such as name, email, or other profile information
- Delete users from the system if required

4. Feedback Management

- View feedback submitted by users
- Displays feedback details including username, email, rating, category, message, and status
- Update feedback status (New, Reviewed, Resolved)

5. Contact Management

- View contact messages sent by users
- Displays details such as date, subject, message, and status
- Manage message status (New, Read, Responded)

6. Logout

- Allows the admin to securely log out of the system

2. User Module :

The User Module is the core learning component of SmartlyGoPHP. It provides users with access to learning content, interactive tools, performance tracking, and communication features.

Sub-Modules & Functionalities :

1. Authentication

- **Sign Up:** New users can register by providing personal and academic details
- **Log In:** Existing users can log in using email and password
- **Log Out:** Securely ends the user session

2. Dashboard

- Displays greeting message
- Shows total score, current level, games played, and daily streak
- Provides quick navigation to tools and learning modules

3. Learning Modules

- Contains structured chapters explaining Core PHP concepts
- Each module includes short descriptions and code examples
- Designed for conceptual clarity and easy understanding

4. Tools & Games

- **Scoring Tools:**
 - Quiz Challenges (MCQ)
 - Fill in the Blanks
 - True / False

Each correct answer increases user score and level progression.
- **Non-Scoring Practice Tools:**
 - Find the Error
 - Fix the Code
 - Flashcards
 - Predict Output
 - Treasure Hunt

These tools focus on practice and logical thinking without affecting scores.

5. Feedback Module

- Users can submit feedback with rating, category, and message

6. Contact Module

- Allows users to send contact messages
- Includes FAQs, email details, contact number, and location information

7. Guest Access

- Users can explore learning content and tools without logging in
- Progress is not saved for guest users

3. Review of Literature

The review of literature focuses on existing learning approaches, online programming platforms, and gamified educational systems that are relevant to the development of SmartlyGoPHP. It highlights how current systems support learning and identifies gaps that the proposed system aims to address.

3.1 Traditional Learning Methods :

Traditional programming education relies heavily on textbooks, classroom lectures, and written exercises. These methods provide strong theoretical foundations but often lack interactivity, instant feedback, and practical exposure. Learners usually depend on instructors for evaluation, which limits self-paced learning and continuous assessment.

3.2 E-Learning Platforms :

Modern e-learning platforms offer online tutorials, video-based learning, and quizzes. While these platforms improve accessibility and flexibility, many of them focus either on theory or assessment, not both together. Additionally, beginner-level learners often find such platforms complex or overwhelming due to advanced content and lack of structured progression.

3.3 Gamification in Learning :

Gamification techniques such as points, levels, badges, and streaks have been widely studied and proven to increase learner motivation and engagement. Educational games help learners practice concepts repeatedly without boredom. However, many systems apply gamification only to quizzes and neglect coding-based problem-solving activities.

3.4 Programming Practice Tools :

Several online tools provide coding challenges, debugging exercises, and output prediction tasks. These tools enhance logical thinking and practical skills but are often scattered across different platforms. Learners need to switch between multiple websites, which reduces learning efficiency and continuity.

3.5 Identified Research Gap :

From the literature review, it is observed that:

- There is a lack of single, integrated platforms that combine learning modules, quizzes, coding practice, and gamification.
- Many systems do not provide progress tracking, accuracy measurement, and streak analysis together.
- Beginner-focused platforms for Core PHP learning with structured practice tools are limited.

4. TECHNICAL DESCRIPTION

4.1 Hardware Requirements :

Since SmartlyGoPHP is developed as a web-based application, it does not require high-end hardware. Any standard computer or laptop capable of running a web browser and local server environment can run the system efficiently.

Minimum Hardware Requirements :

- Processor :** Intel i3 / AMD Ryzen 3 or higher
- RAM :** Minimum 4 GB (8 GB recommended for smoother multitasking)
- Storage :** At least 1 GB free space for project files, server setup, and database
- Display :** Minimum 1366 × 768 screen resolution
- Internet Connection :** Required for accessing the application in hosted mode

Component	Specification
Processor	Intel i3 / AMD Ryzen 3 or higher
RAM	Minimum 4 GB (8 GB recommended)
Storage	At least 1 GB free space
Display	Minimum 1366 × 768 resolution
Internet	Required for hosted deployment

Table 3.1 Hardware Configuration

4.2 Software Requirements :

The software environment for SmartlyGoPHP is based on open-source and widely used web technologies, ensuring cost-effectiveness, flexibility, and easy deployment.

Software Stack Used :

- Operating System** : Windows 10 / 11 (Compatible with Linux and macOS)
- Programming Language** : PHP 8.0 or higher (Core PHP)
- Frontend Technologies** : HTML5, CSS3, JavaScript
- Database** : MySQL (using phpMyAdmin for management)
- Web Server** : Apache Server (via XAMPP / WAMP / LAMP)
- Browser** : Google Chrome, Mozilla Firefox, or any modern browser
- IDE / Code Editor** : VS Code / Sublime Text / PHPStorm

Software Component	Specification
Operating System	Windows 10 / 11 (Linux & macOS supported)
Programming Language	PHP 8.0 or higher
Frontend	HTML, CSS, JavaScript
Database	MySQL
Web Server	Apache (XAMPP/WAMP/LAMP)
IDE / Editor	VS Code / Sublime Text / PHPStorm
Browser	Chrome / Firefox / Edge

Table 3.2 Software Configuration

Conclusion of Hardware and Software requirements :

The system requirements of SmartlyGoPHP are minimal and practical, making it suitable for deployment on standard systems commonly available to students and institutions. This ensures easy installation, execution, and scalability.

5. SYSTEM DESIGN AND DEVELOPMENT

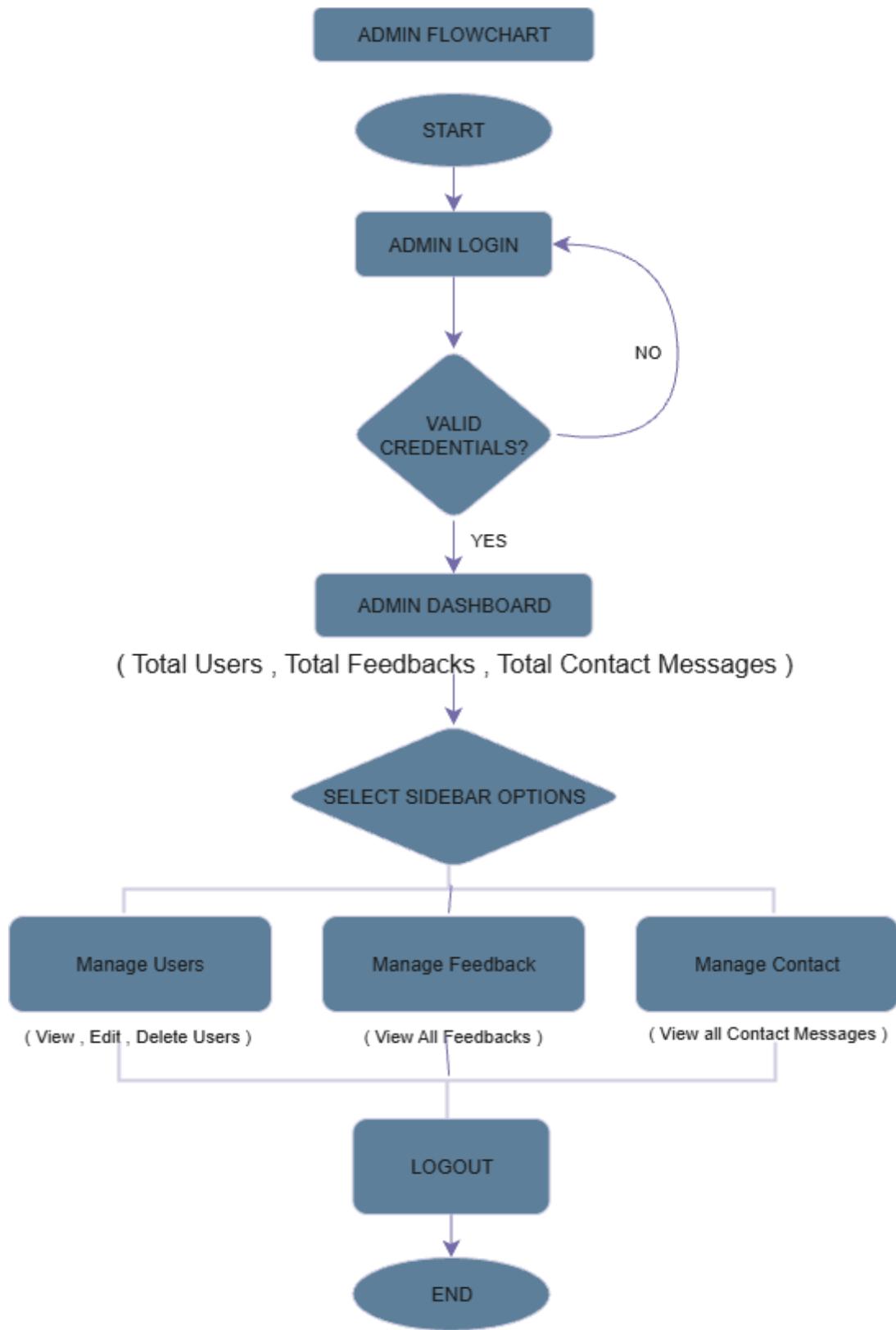


Figure 5.1 Admin Flow Chart

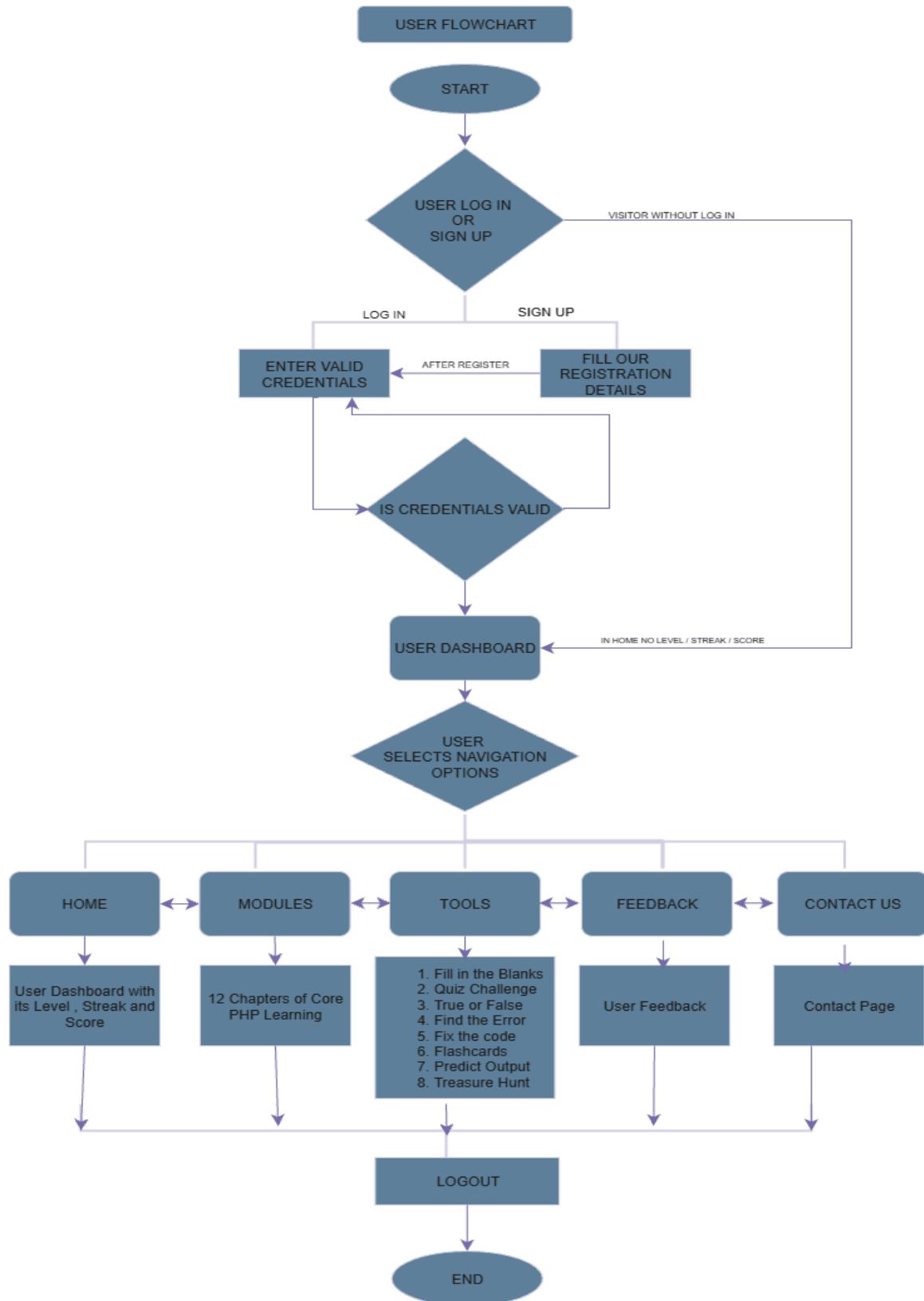


Figure 5.1.2 User Flow Chart

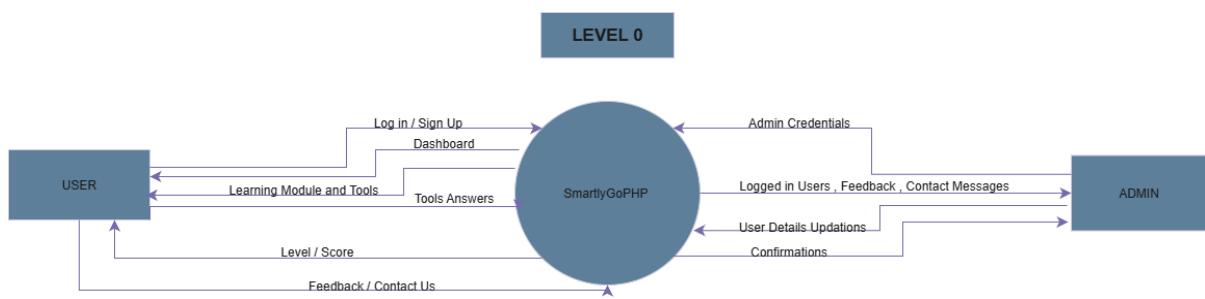


Figure 5.2 DFD Level 0

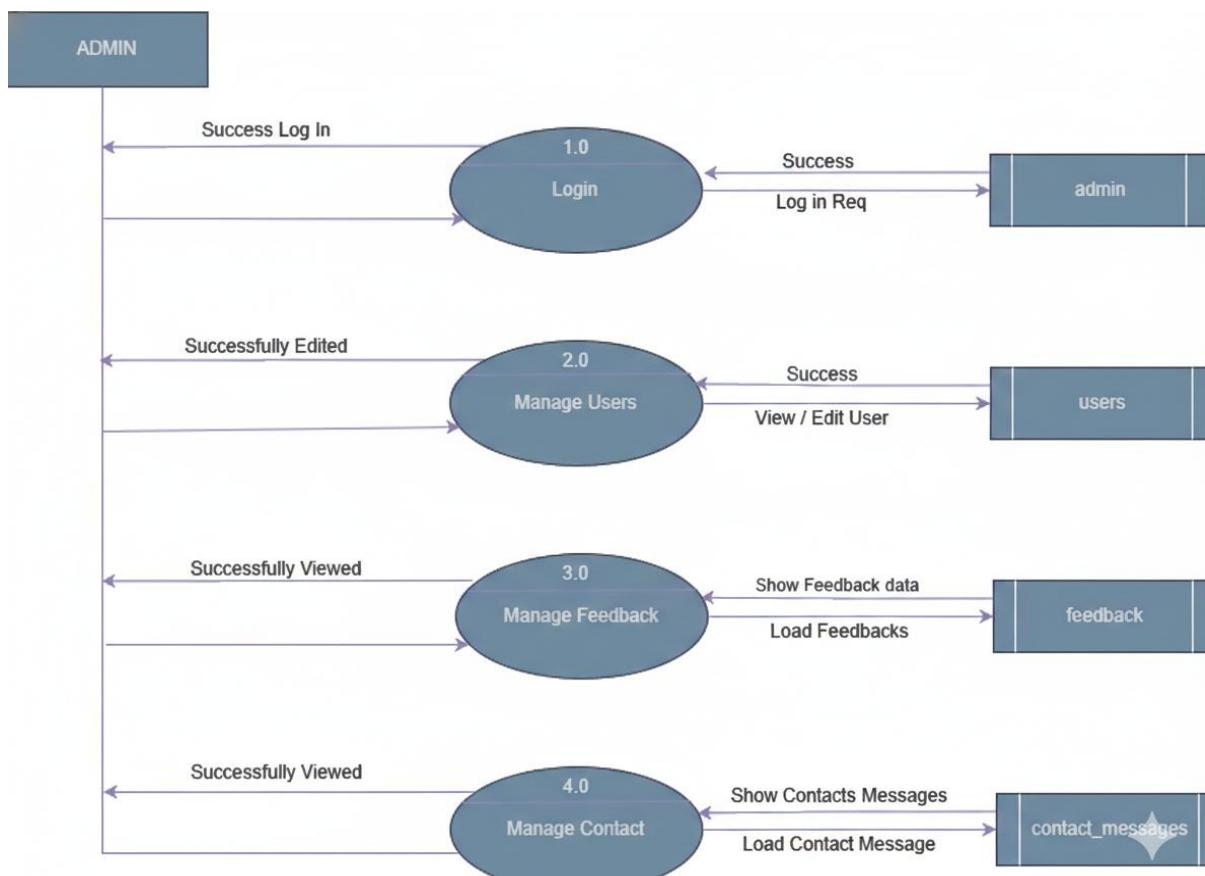


Figure 5.2.1 DFD Level 1 Admin

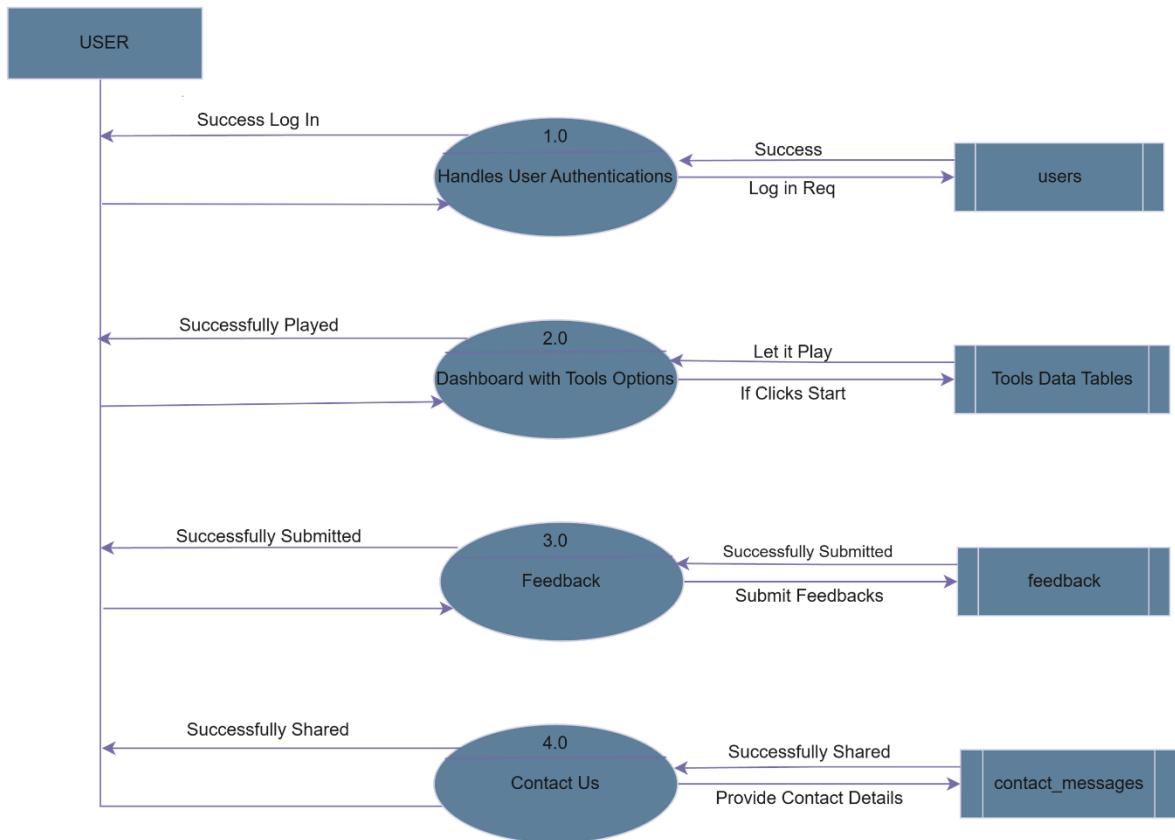


Figure 5.2.3 DFD Level 1 User

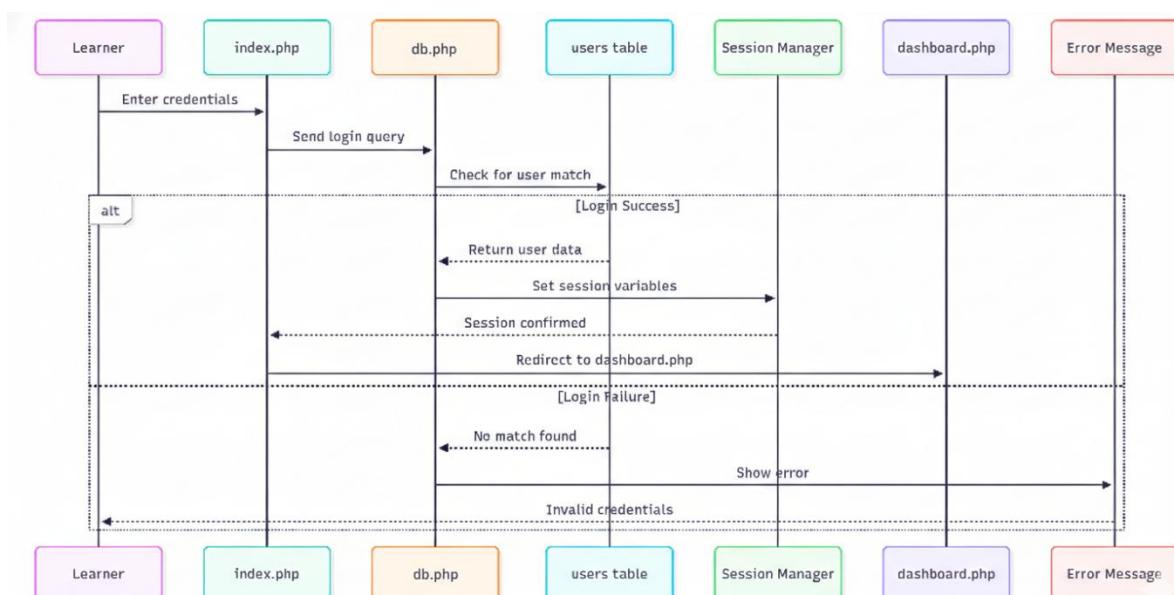


Figure 5.3 Sequential Diagram of Authentication

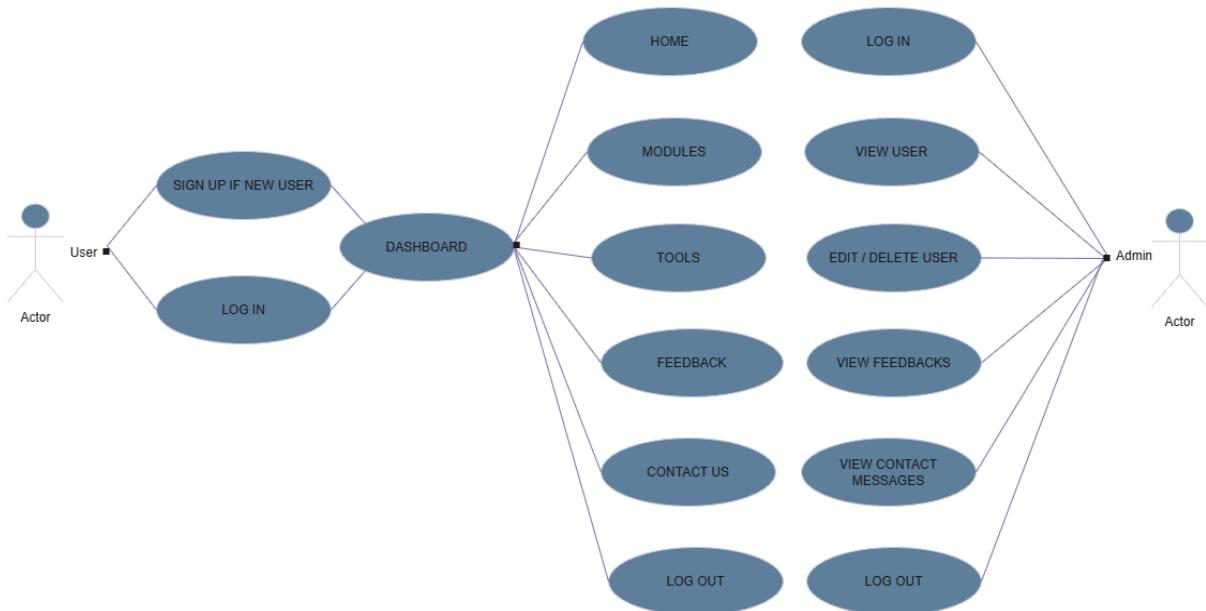


Figure 5.4 Use Case Diagram

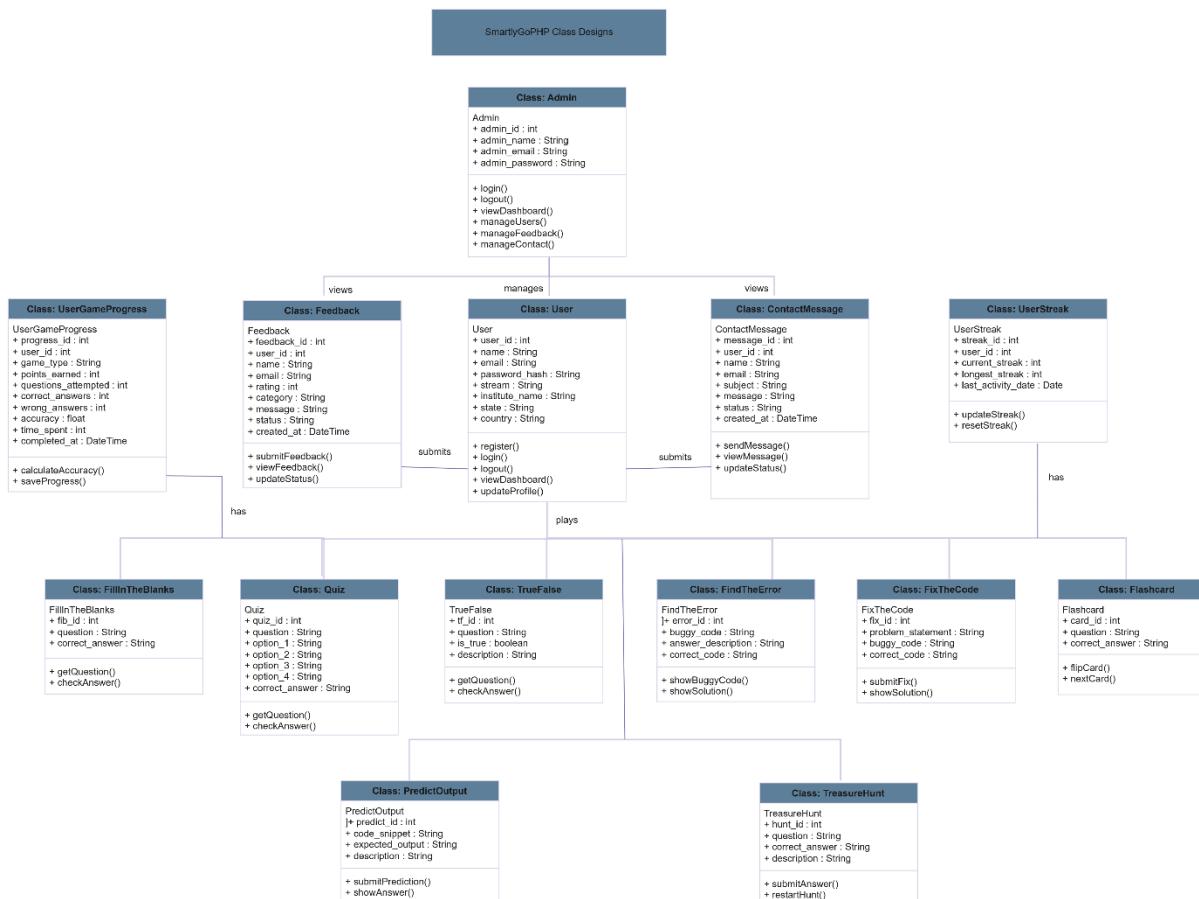


Figure 5.5 Class Diagram

5.6 DATABASE DESIGN

DATABASE TABLE DESCRIPTION :

The **SmartlyGo_db** database is designed using a relational model to efficiently store user information, learning content, game data, progress tracking, and administrative records. Each table has a specific role in supporting the overall functionality of the SmartlyGoPHP learning platform.

1. Admin Table :

The admin table stores login credentials and details of administrators who manage the system.

Purpose:

- Used for secure admin authentication.
- Grants access to the admin dashboard and management features.

Attributes Description:

- admin_id: Unique identifier for each admin (Primary Key).
- admin_name: Name of the administrator.
- admin_email: Email ID of the admin (used for login).
- admin_password: Encrypted password for secure authentication.

2. Users Table :

The users table stores personal and academic details of registered users.

Purpose:

- Manages user accounts.
- Used for authentication and personalization.

Attributes Description:

- user_id: Unique identifier for each user (Primary Key).
- name: Full name of the user.
- email: User email address (used for login).
- password_hash: Encrypted password.
- stream: Academic stream of the user.
- institute_name: Name of the institution.
- state: State of the user.
- country: Country of the user.

3. Feedback Table :

The feedback table stores feedback submitted by users regarding the platform.

Purpose:

- Helps administrators analyze user satisfaction.
- Improves platform quality based on user input.

Attributes Description:

- feedback_id: Unique feedback identifier (Primary Key).
- user_id: References the user who submitted feedback.
- name: Name of the user.
- email: Email of the user.
- rating: Star rating given by the user.
- category: Type of feedback (content, UI, performance, etc.).
- message: Feedback description.
- status: Feedback status (New, Reviewed, Resolved).
- created_at: Timestamp of submission.

4. Contact_Message Table :

The contact message table stores messages sent through the Contact Us form.

Purpose:

- Handles user queries and communication.

Attributes Description:

- message_id: Unique message identifier (Primary Key).
- user_id: User reference (nullable for guest users).
- name: Sender name.
- email: Sender email address.
- subject: Subject of the message.
- message: Message content.
- status: Message status (New, Read, Responded).
- created_at: Date and time of submission.

5. Quiz Table :

The quiz table stores MCQ-based questions.

Purpose:

- Used in quiz challenges for score-based learning.

Attributes Description:

- quiz_id: Unique question identifier (Primary Key).
- question: Quiz question text.
- option_1 to option_4: Multiple choice options.
- correct_answer: Correct option value.

6. FillInTheBlanks Table :

The fillintheblanks table stores fill-in-the-blank questions.

Purpose:

- Tests conceptual understanding of PHP syntax and logic.

Attributes Description:

- fib_id: Unique question identifier (Primary Key).
- question: Sentence with missing word.
- correct_answer: Correct missing value.

7. TrueFalse Table :

The truefalse table stores true/false questions.

Purpose:

- Evaluates basic conceptual clarity.

Attributes Description:

- tf_id: Unique identifier (Primary Key).
- question: Statement to evaluate.
- is_true: Boolean value indicating correct answer.
- description: Explanation of the answer.

8. FindTheError Table :

The findtheerror table stores buggy code samples.

Purpose:

- Enhances debugging skills.

Attributes Description:

- error_id: Unique identifier (Primary Key).
- buggy_code: Code containing error.
- answer_description: Explanation of the error.
- correct_code: Corrected version of the code.

9. FixTheCode Table :

The fixthecode table stores code correction challenges.

Purpose:

- Improves hands-on coding and correction skills.

Attributes Description:

- fix_id: Unique identifier (Primary Key).
- problem_statement: Description of the issue.
- buggy_code: Incorrect code.
- correct_code: Correct solution code.

10. FlashcardFlipper Table :

The flashcardflipper table stores flashcard questions.

Purpose:

- Used for quick revision and concept recall.

Attributes Description:

- card_id: Unique card identifier (Primary Key).
- question: Flashcard question.
- correct_answer: Answer shown on card flip.

11. PredictOutput Table :

The **predictoutput** table stores code-based output prediction questions.

Purpose:

- Strengthens logical reasoning and output understanding.

Attributes Description:

- predict_id: Unique identifier (Primary Key).
- code_snippet: Code provided to user.
- expected_output: Correct output.
- description: Explanation of output logic.

12. TreasureHunt Table :

The **treasurehunt** table stores clue-based questions.

Purpose:

- Encourages logical thinking through puzzles.

Attributes Description:

- hunt_id: Unique identifier (Primary Key).
- question: Clue or puzzle.
- correct_answer: Correct solution.
- description: Additional explanation or hint.

13. User_Game_Progress Table :

The **user_game_progress** table tracks user performance.

Purpose:

- Maintains scoring, accuracy, and activity logs.

Attributes Description:

- progress_id: Unique progress record (Primary Key).
- user_id: Reference to user.
- game_type: Type of game played.
- points_earned: Points scored.
- questions_attempted: Total attempts.
- correct_answers: Correct responses.
- wrong_answers: Incorrect responses.
- accuracy: Percentage accuracy.
- time_spent: Time spent in seconds.
- completed_at: Date and time of activity.

14. User_Streaks Table :

The user streaks table tracks learning consistency.

Purpose:

- Encourages daily engagement.

Attributes Description:

- streak_id: Unique streak identifier (Primary Key).
- user_id: Reference to user.
- current_streak: Ongoing streak count.
- longest_streak: Highest streak achieved.
- last_activity_date: Date of last activity.

Conclusion of DB Design Schema :

The database design of **SmartlyGoPHP** is comprehensive, normalized, and scalable. Each table plays a vital role in ensuring smooth learning, accurate progress tracking, secure administration, and interactive user experience.

5.7 Screen Design & Coding

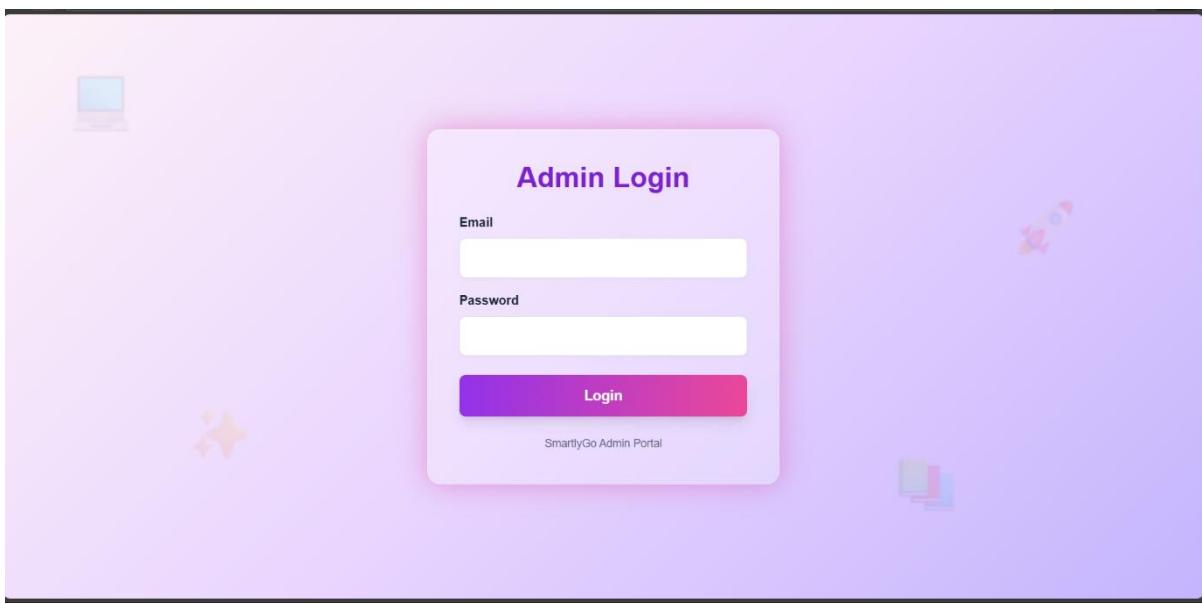


Figure 5.7.1 Admin Login Page

```
<?php
session_start();
require_once 'db.php';

if (isset($_POST['login_btn'])) {
    $email = trim($_POST['email']);
    $password = trim($_POST['password']);
    // Fetch admin
    $sql = "SELECT * FROM admin WHERE admin_email = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 1) {
        $admin = $result->fetch_assoc();

        // Compare MD5 password
        if ($admin['admin_password'] === md5($password)) {
            $_SESSION['admin_id'] = $admin['admin_id'];
            $_SESSION['admin_name'] = $admin['admin_name'];

            header("Location: admin-dashboard.php");
            exit;
        } else {
            $error = "Invalid password";
        }
    } else {
        $error = "Admin not found";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        body {
            background: linear-gradient(135deg, #fdf2f8, #e9d5ff, #c4b5fd);
            height: 100vh;
            overflow: hidden;
            font-family: 'Poppins', sans-serif;
        }
    </style>

```

```

/* Floating Animated Objects */
.float-object {
    position: absolute;
    font-size: 70px;
    opacity: 0.15;
    animation: float 8s ease-in-out infinite;
    filter: blur(1px);
    user-select: none;
}
.obj1 { top: 10%; left: 5%; animation-delay: 0s; }
.obj2 { top: 65%; left: 15%; animation-delay: 1.5s; }
.obj3 { top: 30%; right: 10%; animation-delay: 3s; }
.obj4 { bottom: 10%; right: 20%; animation-delay: 4.5s; }

@keyframes float {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-25px); }
}

/* Card Glow Animation */
.login-card {
    backdrop-filter: blur(15px);
    background: rgba(255, 255, 255, 0.25);
    border: 1px solid rgba(255, 255, 255, 0.3);
    animation: glow 4s infinite alternate;
}

@keyframes glow {
    from { box-shadow: 0 0 20px rgba(168, 85, 247, 0.4); }
    to { box-shadow: 0 0 40px rgba(236, 72, 153, 0.5); }
}

</style>
</head>
<body class="flex items-center justify-center relative">
    <!-- Floating Emojis -->
    <div class="float-object obj1">💻</div>
    <div class="float-object obj2">💡</div>
    <div class="float-object obj3">🔖</div>
    <div class="float-object obj4">📊</div>

    <!-- LOGIN CARD -->
    <div class="login-card w-full max-w-md p-10 rounded-2xl shadow-2xl relative z-20">

        <h2 class="text-4xl font-bold text-center mb-6 text-purple-700 drop-shadow-sm">
            Admin Login
        </h2>

        <?php if (isset($error)): ?>
        <div class="bg-red-100 text-red-700 p-3 rounded mb-4 text-center font-semibold shadow">
            <?php echo $error; ?>
        </div>
        <?php endif; ?>

        <form method="POST">
            <label class="block mb-2 font-semibold text-gray-800">Email</label>
            <input type="email" name="email" required
                class="w-full p-3 mb-4 rounded-lg border outline-none focus:ring-2 focus:ring-purple-500 shadow-sm">

            <label class="block mb-2 font-semibold text-gray-800">Password</label>
            <input type="password" name="password" required
                class="w-full p-3 mb-6 rounded-lg border outline-none focus:ring-2 focus:ring-purple-500 shadow-sm">

            <button type="submit" name="login_btn"
                class="w-full bg-gradient-to-r from-purple-600 to-pink-500 text-white p-3 text-lg font-semibold rounded-lg shadow-lg
                hover:opacity-90 transition">
                Login
            </button>
        </form>

        <!-- Small footer -->
        <p class="text-center mt-6 text-gray-700 font-medium text-sm opacity-75">
            SmartlyGo Admin Portal
        </p>
    </div>
</body>
</html>

```

Code 5.7.2 admin-login.php

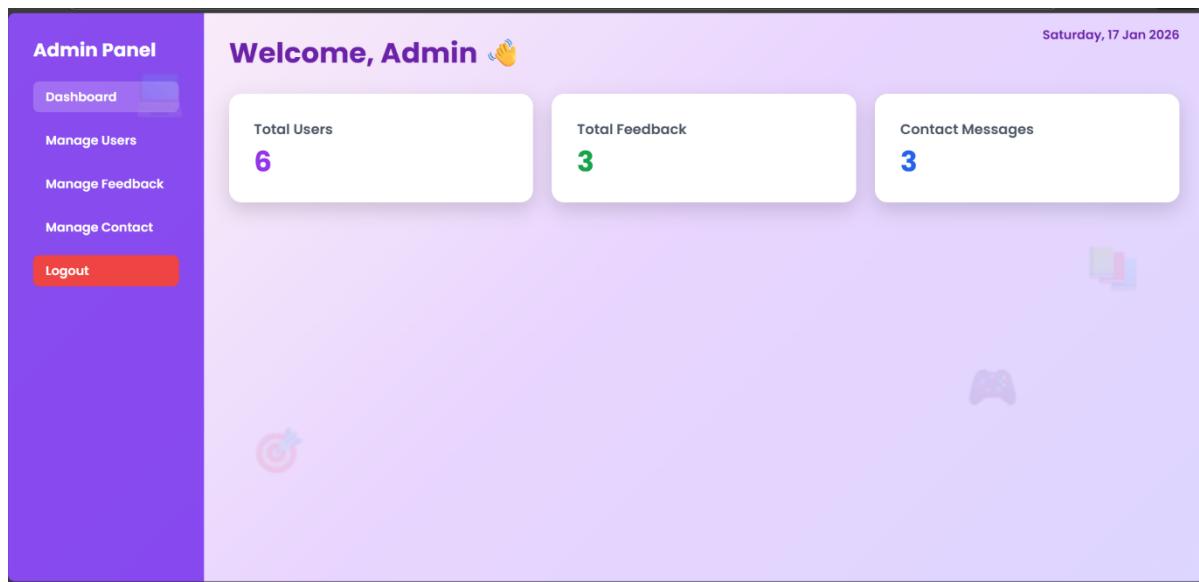


Figure 5.7.3 Admin Dashboard Page

#	Name	Email	Stream	Institute	State	Country	Action
1	Het Parmar	het@gmail.com	Arts	Marwadi University	Gujarat	INDIA	<button>Edit</button> <button>Delete</button>
2	Vipul Bambhaniya	vipul23@gmail.com	Commerce	Marwadi University	Gujarat	INDIA	<button>Edit</button> <button>Delete</button>
3	Harsh Dodiya	harshl@gmail.com	Science	Jk College	Gujarat	India	<button>Edit</button> <button>Delete</button>
4	Harsh Dodiya	harsh@gmail.com	Science	Jk College	Gujarat	India	<button>Edit</button> <button>Delete</button>
5	Jenish Gohel	jenishgohel@gmail.com	Commerce	Marwadi University	Gujarat	India	<button>Edit</button> <button>Delete</button>
6	Vipul Bambhaniya	vipul@gmail.com	Commerce	Marwadi University	Gujarati	India	<button>Edit</button> <button>Delete</button>
7	Ravi Sav	ravi@gmail.com	BCA	Darshan University	Gujarat	India	<button>Edit</button> <button>Delete</button>

Figure 5.7.4 Manage Users Page

#	Name	Email	Stream	Institute	State	Country	Action
1	Het Parmar	het@gmail.com	Arts	Marwadi University	Gujarat	INDIA	<button>Edit</button> <button>Delete</button>
2	Vipul Bambhaniya	vipull23@gmail.com	Commerce	Marwadi University	Gujarat	INDIA	<button>Edit</button> <button>Delete</button>
3	Harsh Dodiya	harsh@gmail.com	Science	Jk College	Gujarat	India	<button>Edit</button> <button>Delete</button>
4	Jenish Gohel	jenishgohel@gmail.com	Commerce	Marwadi University	Gujarat	India	<button>Edit</button> <button>Delete</button>
5	Vipul Bambhaniya	vipul@gmail.com	Commerce	Marwadi University	Gujarati	India	<button>Edit</button> <button>Delete</button>
6	Ravi Sav	ravi@gmail.com	BCA	Darshan University	Gujarat	India	<button>Edit</button> <button>Delete</button>

Figure 5.7.5 Admin Edit / Delete User Options

Edit User Details

Full Name

Stream

Institute Name

State

Country

Update User
Cancel

Figure 5.7.6 Edit User Details Page

Manage Feedback								Friday, 30 Jan 2026
#	Name	Email	Category	Rating	Status	Message	Date	
1	Harsh Dodiya	harsh@gmail.com	Games	5/5	New	Very Nice Challenges for core concepts	17 Jan 2026	
2	Jenish Gohel	jenishgohel@gmail.com	Content	4/5	New	Keep It UP. Good Concept with attractive UI.	16 Jan 2026	
3	Vipul Bambhaniya	vipul@gmail.com	Games	5/5	New	Appreciable	09 Jan 2026	

Figure 5.7.7 Manage Feedback Page

Manage Contact Messages								Friday, 30 Jan 2026
#	Name	Email	Subject	Message	Status	Date		
1	Ravi Sav	ravi@gmail.com	Regarding Contact	NA	New	16 Jan 2026		
2	Vipul Bambhaniya	vipul@gmail.com	Appreciation	This site is really good for learning.	New	09 Jan 2026		
3	Jenish Gohel	jenishgohell@gmail.com	Relevant to Games	Its Very Nice to learn Core PHP Concepts	New	15 Dec 2025		

Figure 5.7.8 Manage Contact Page

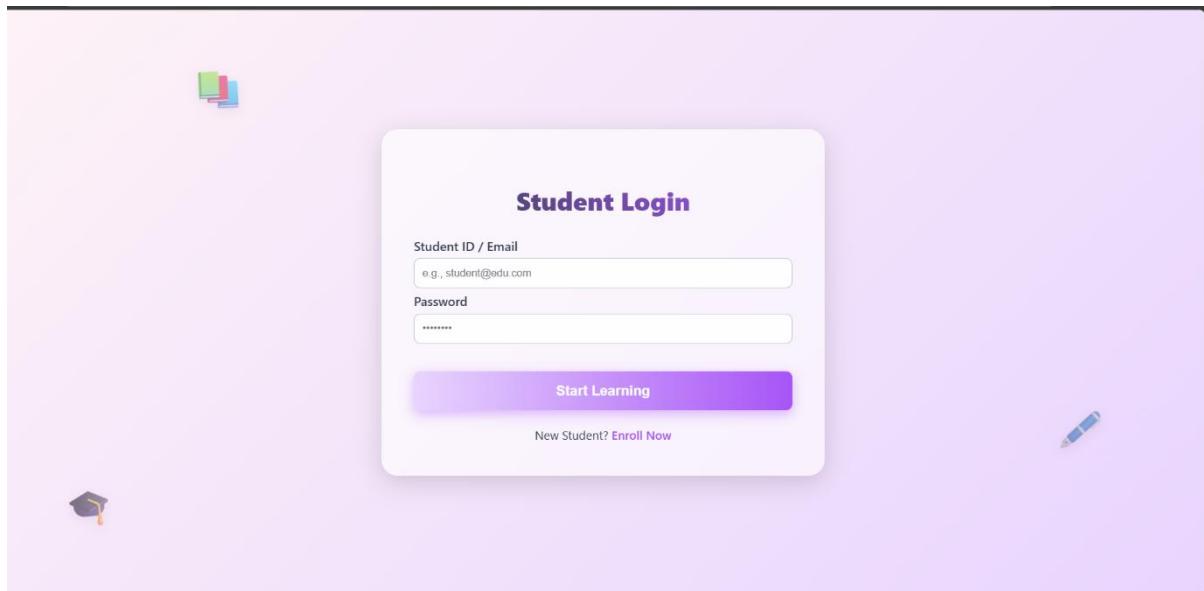


Figure 5.7.9 Student Login Page

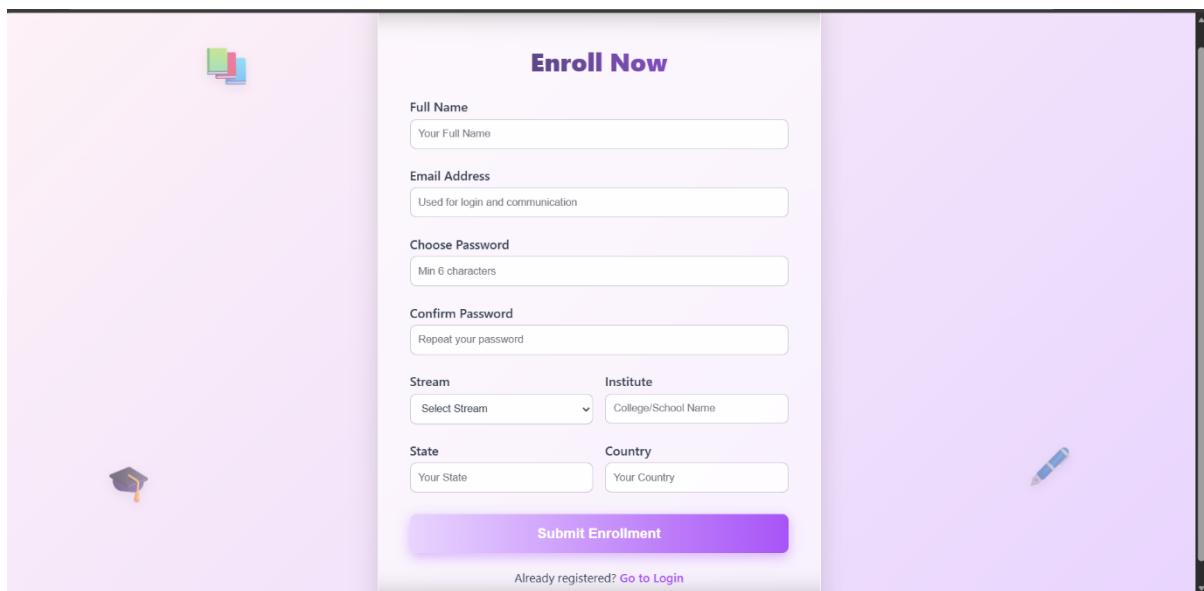


Figure 5.7.10 Student Registration Page

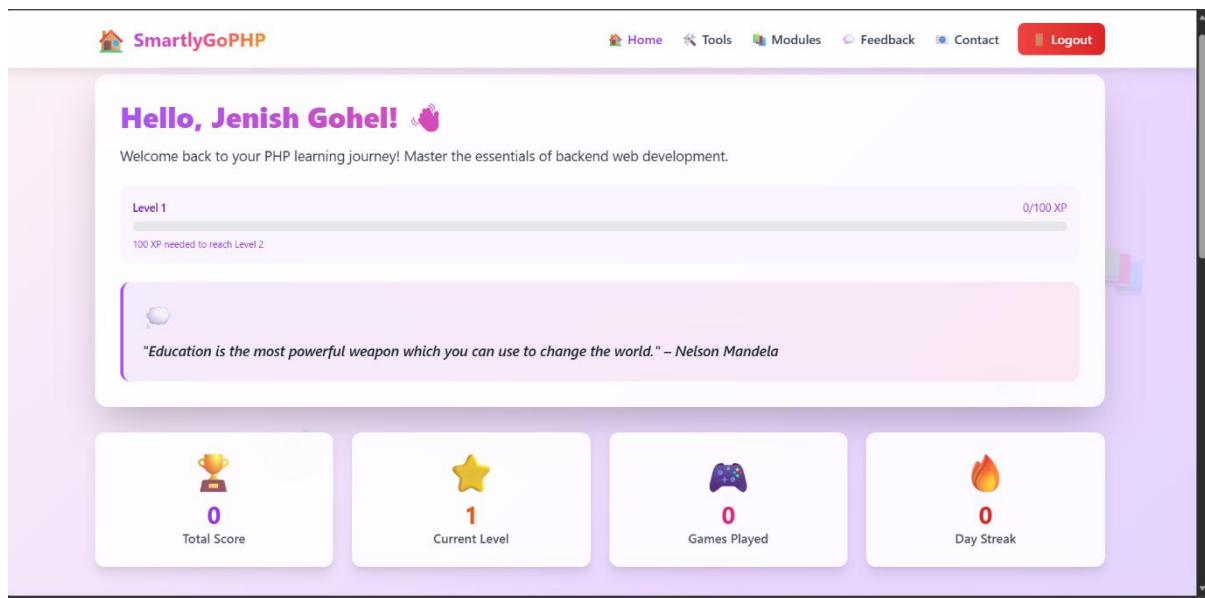


Figure 5.7.11 User Dashboard Page 1

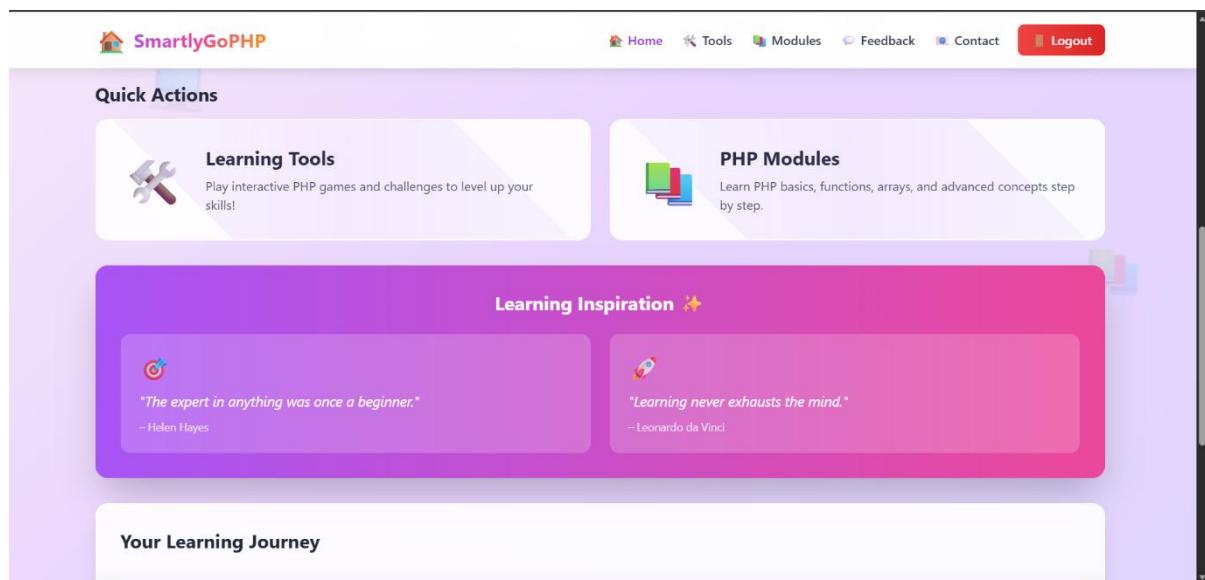


Figure 5.7.12 User Dashboard Page 2

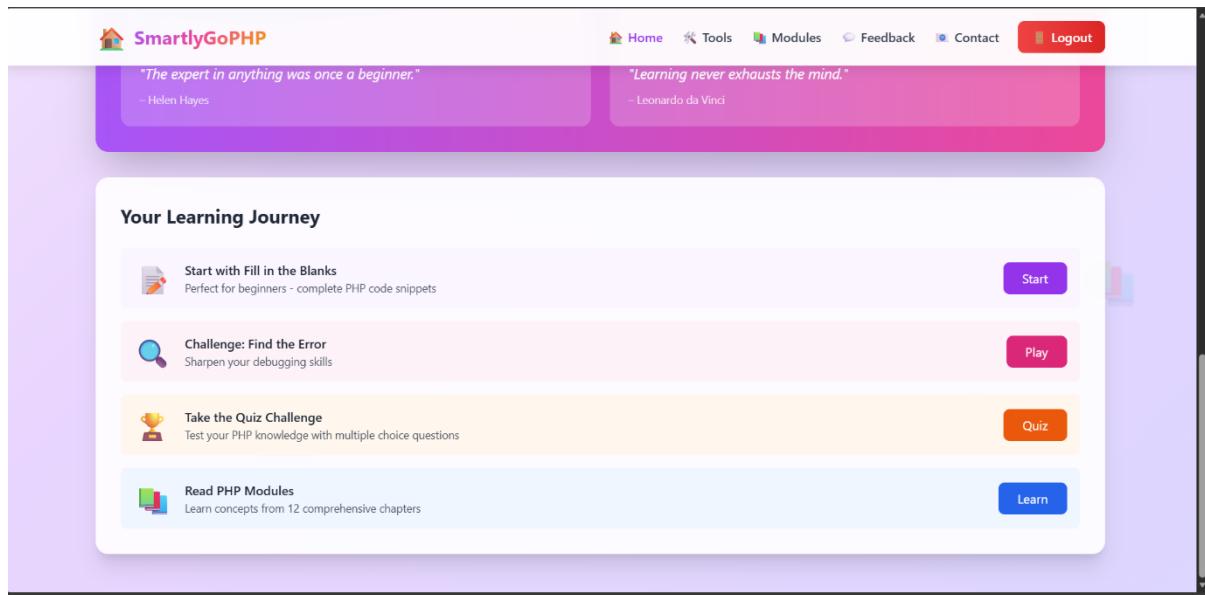


Figure 5.7.13 User Dashboard Page 3

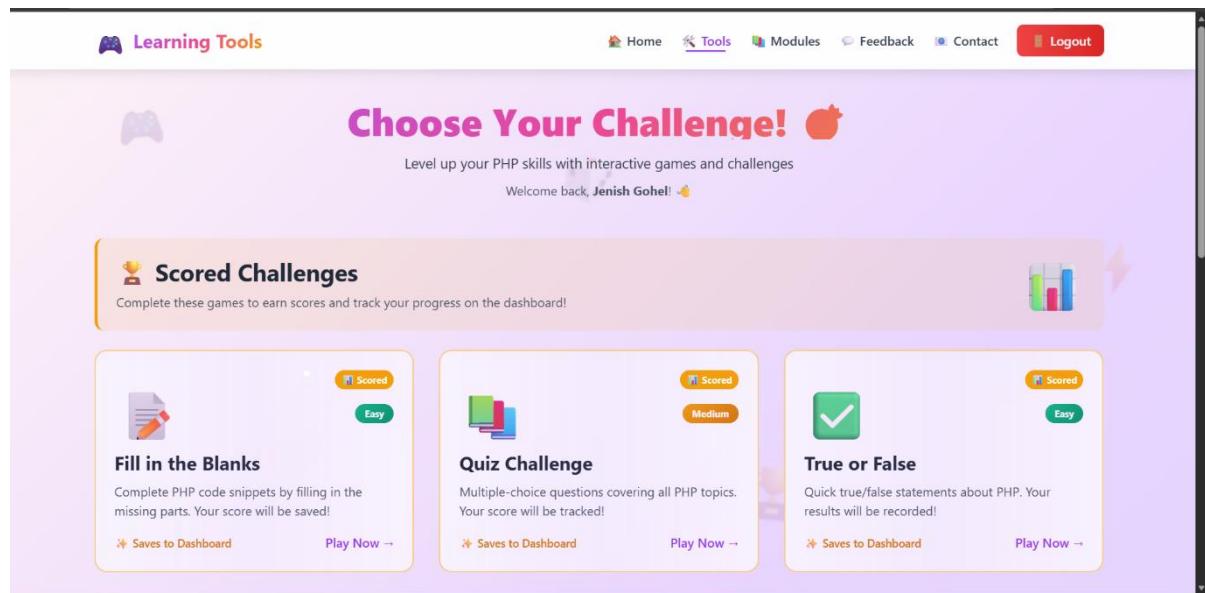


Figure 5.7.14 Tools Page 1

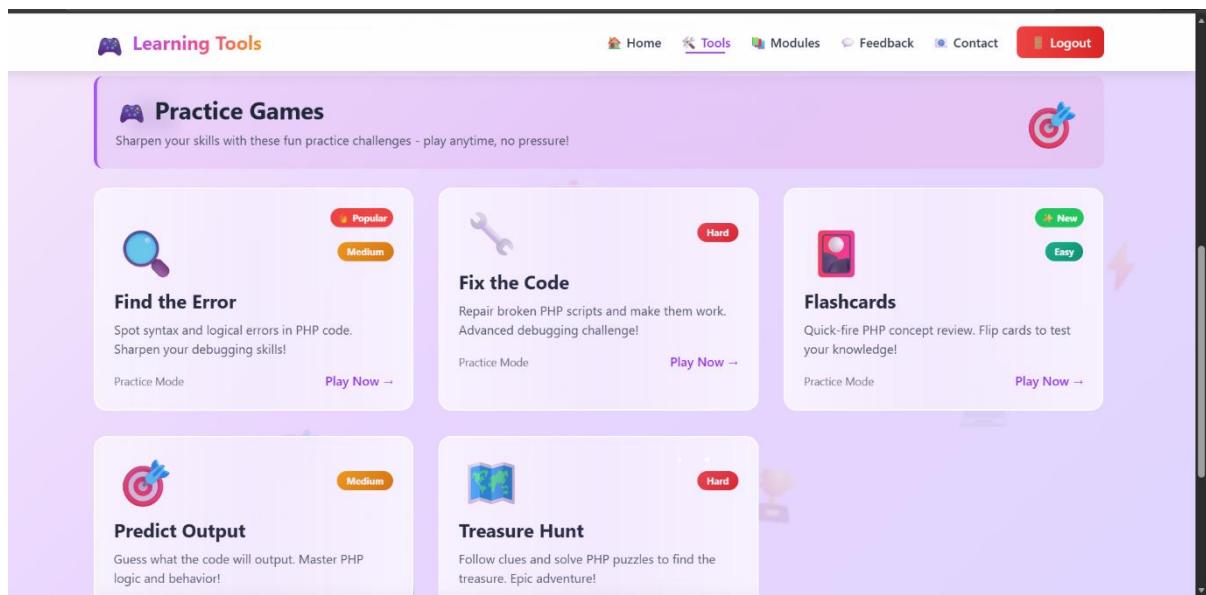


Figure 5.7.15 Tools Page 2

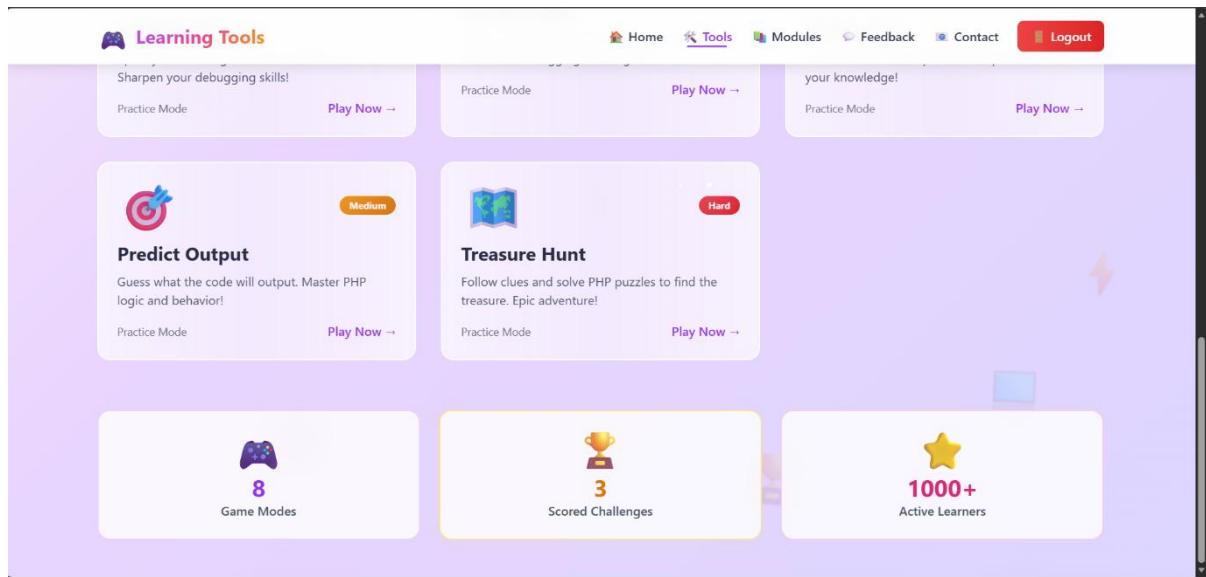


Figure 5.7.16 Tools Page 3

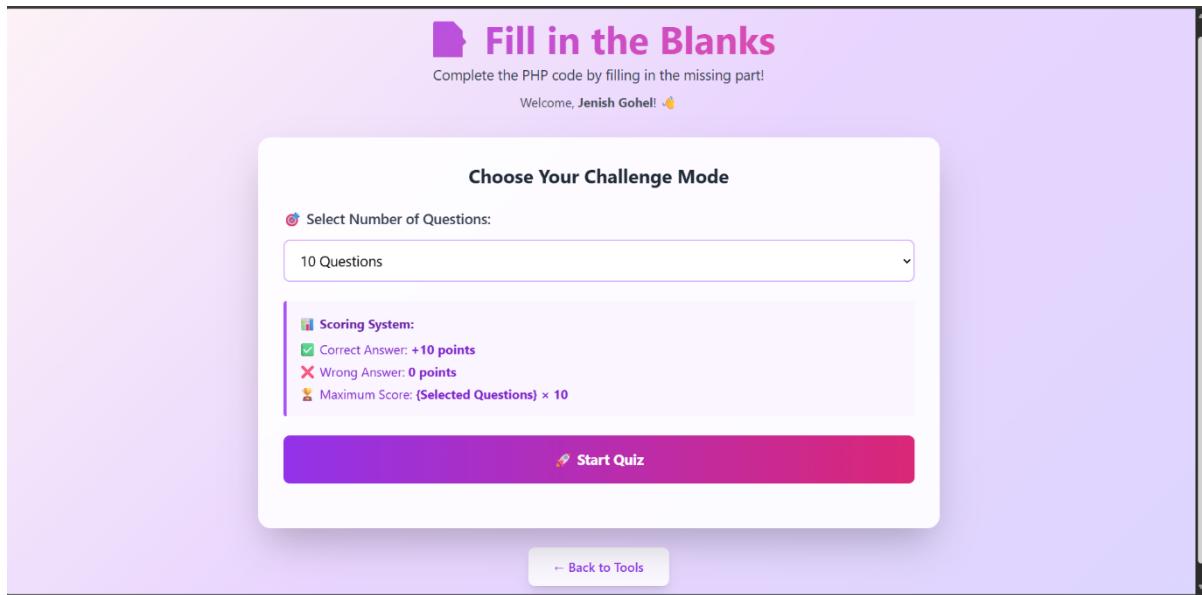


Figure 5.7.17 Fill in the Blanks Page 1

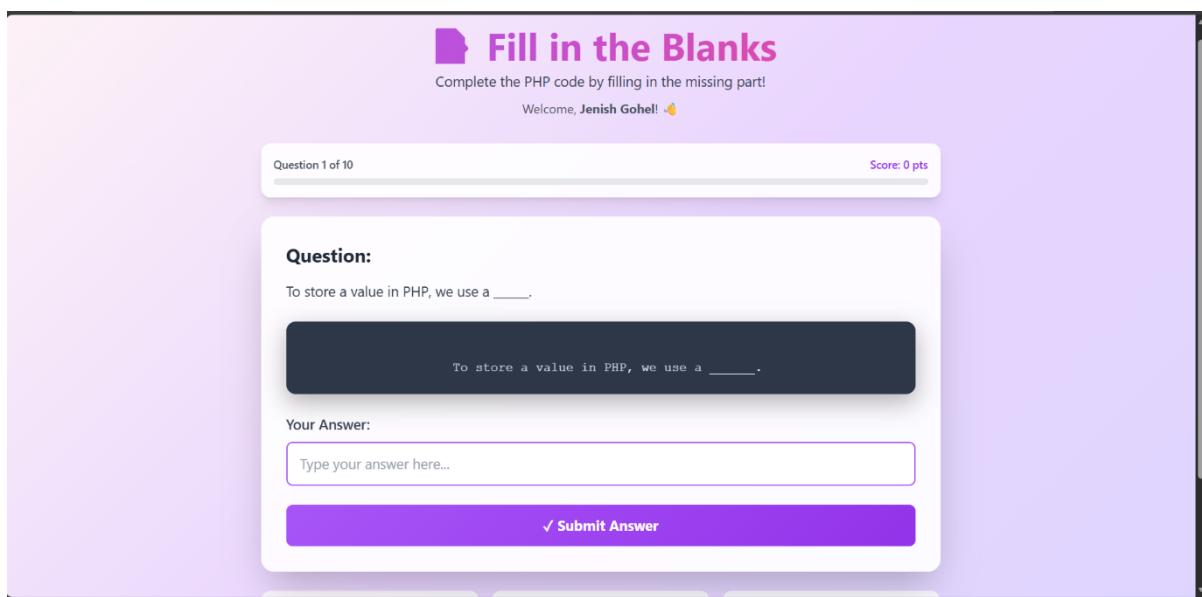


Figure 5.7.18 Fill in the Blanks Page 2

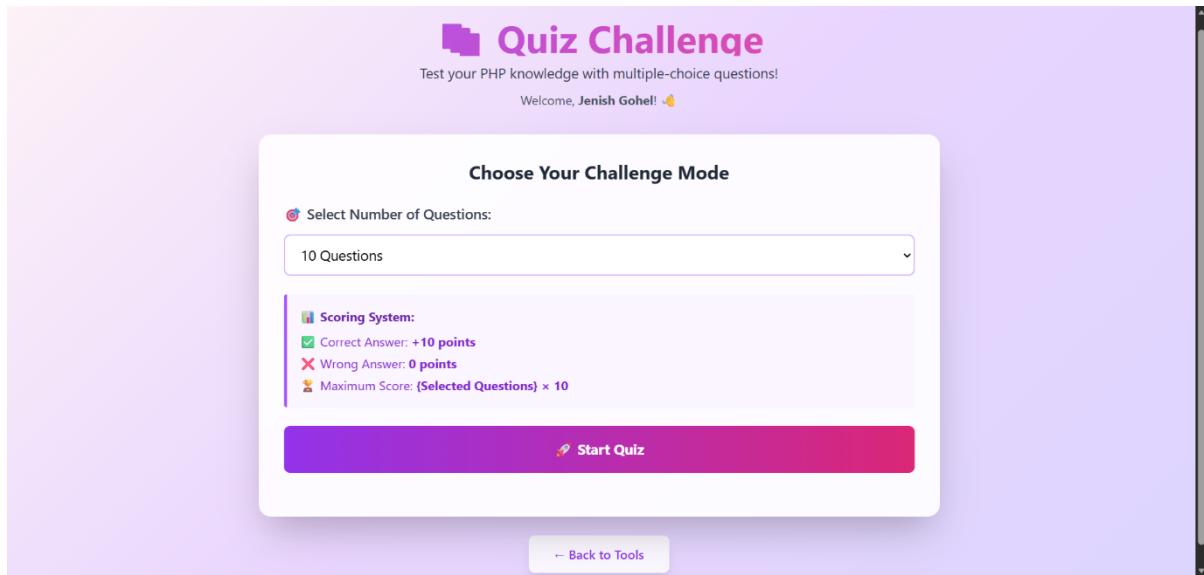


Figure 5.7.19 Quiz Challenge Page 1

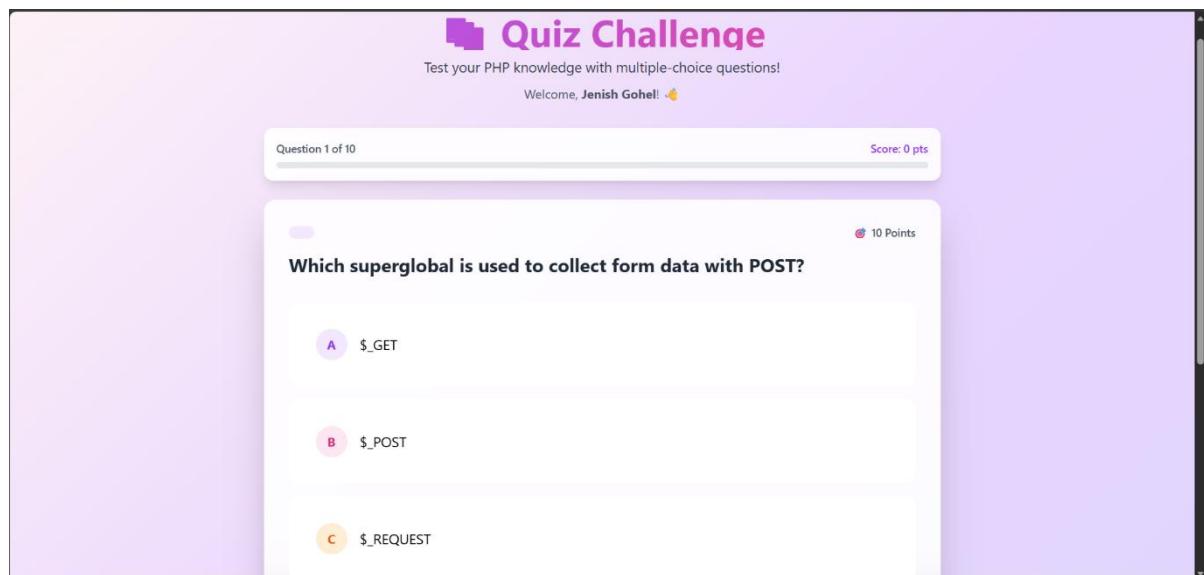


Figure 5.7.20 Quiz Challenge Page 2

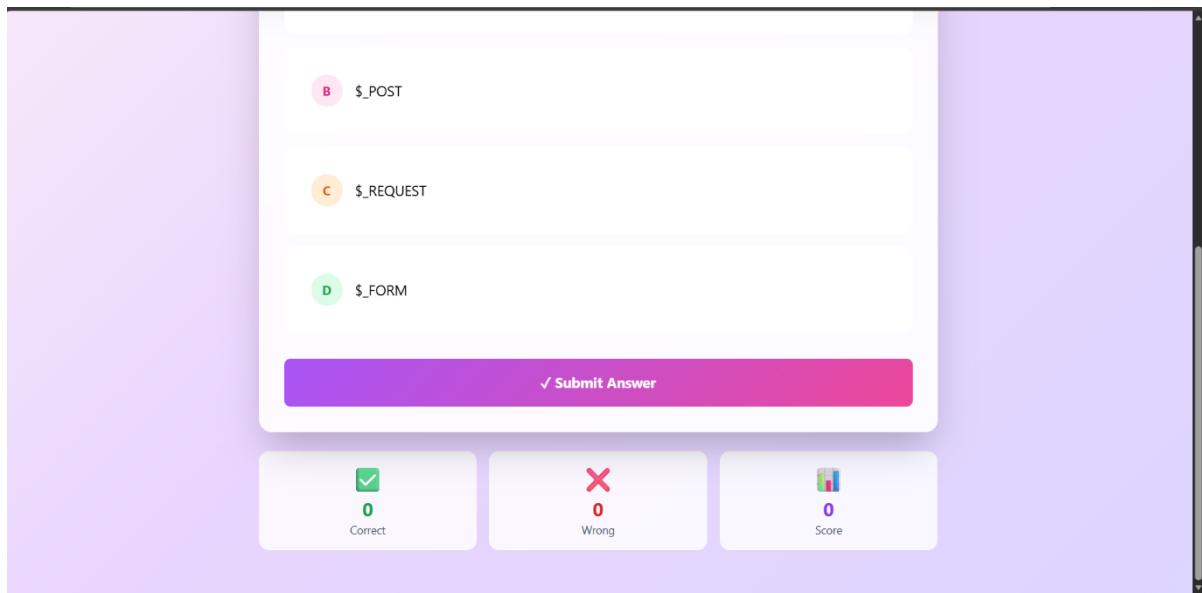


Figure 5.7.21 Quiz Challenge Page 3

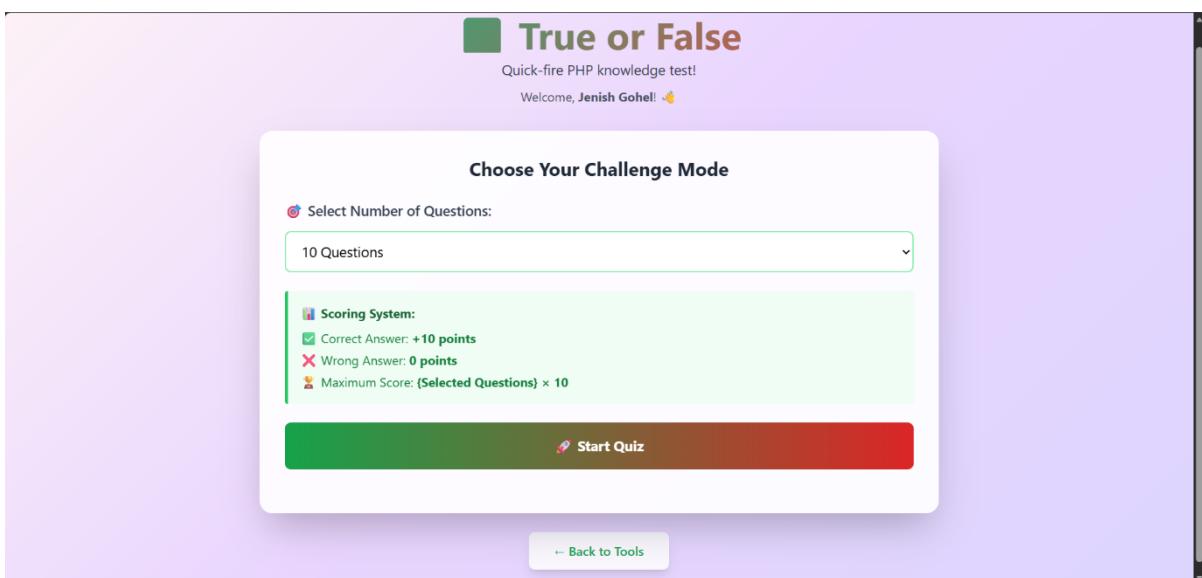


Figure 5.7.22 True or False Page 1

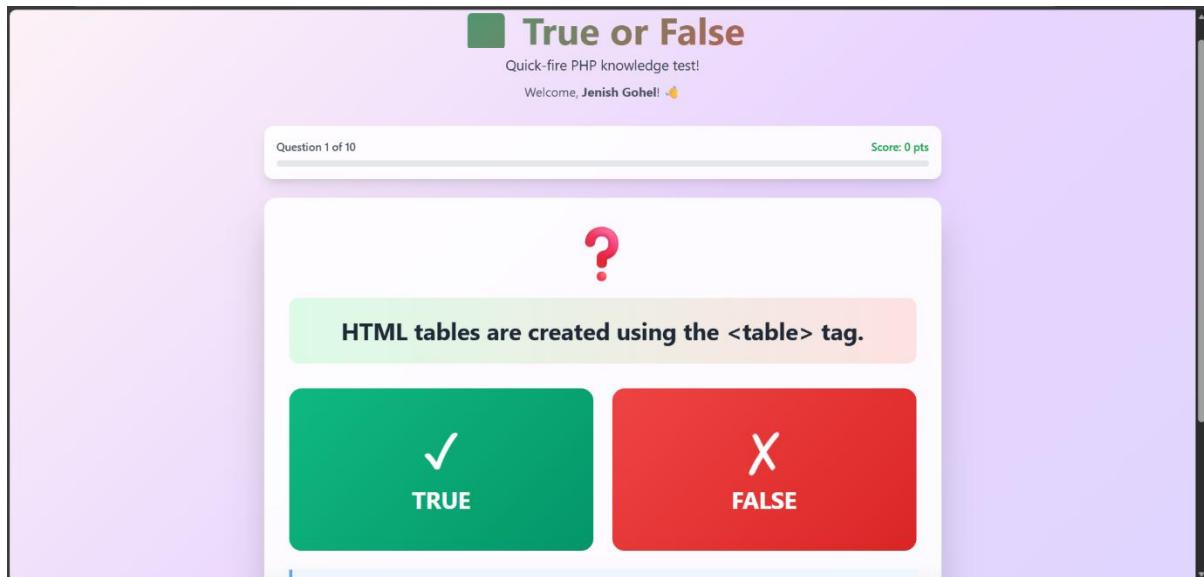


Figure 5.7.23 True or False Page 2

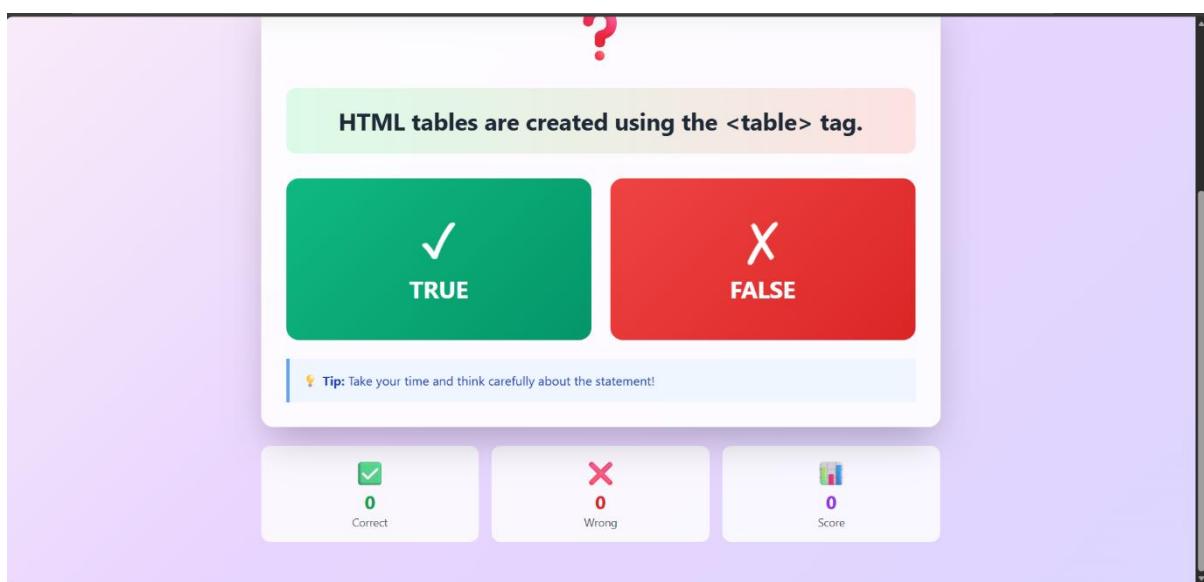


Figure 5.7.24 True or False Page 3

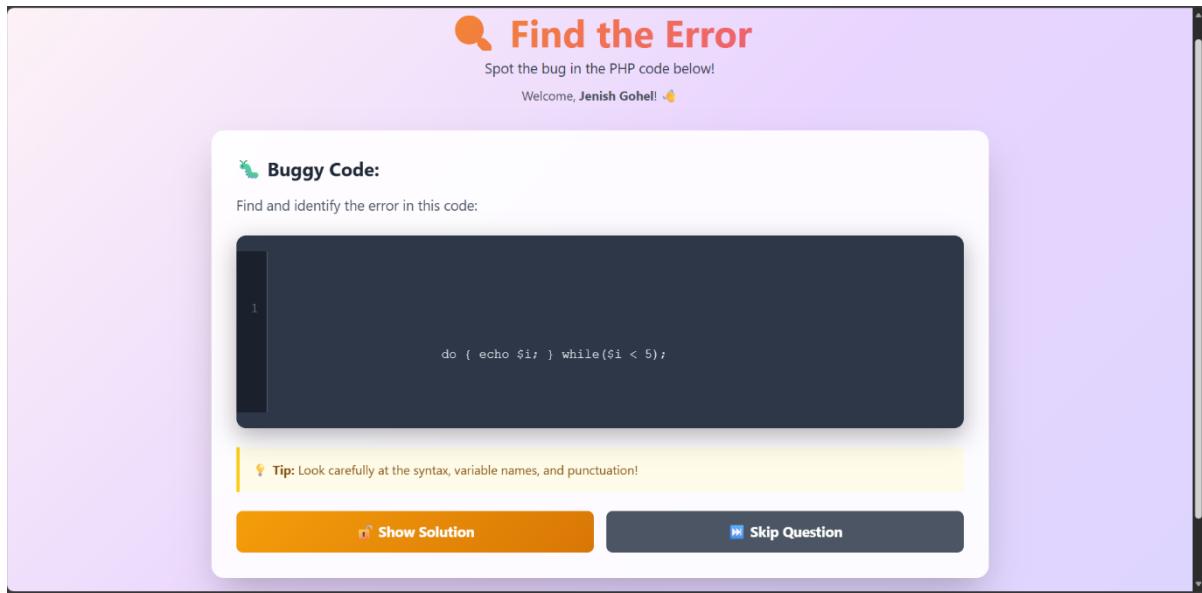


Figure 5.7.25 Find the Error Page

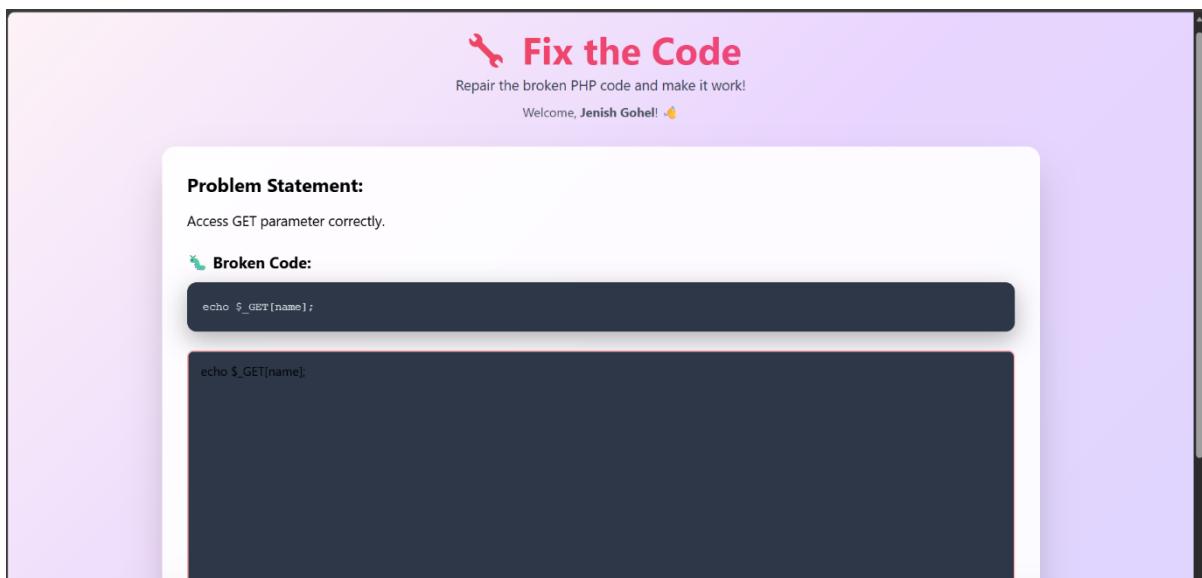


Figure 5.7.26 Fix the Code Page 1

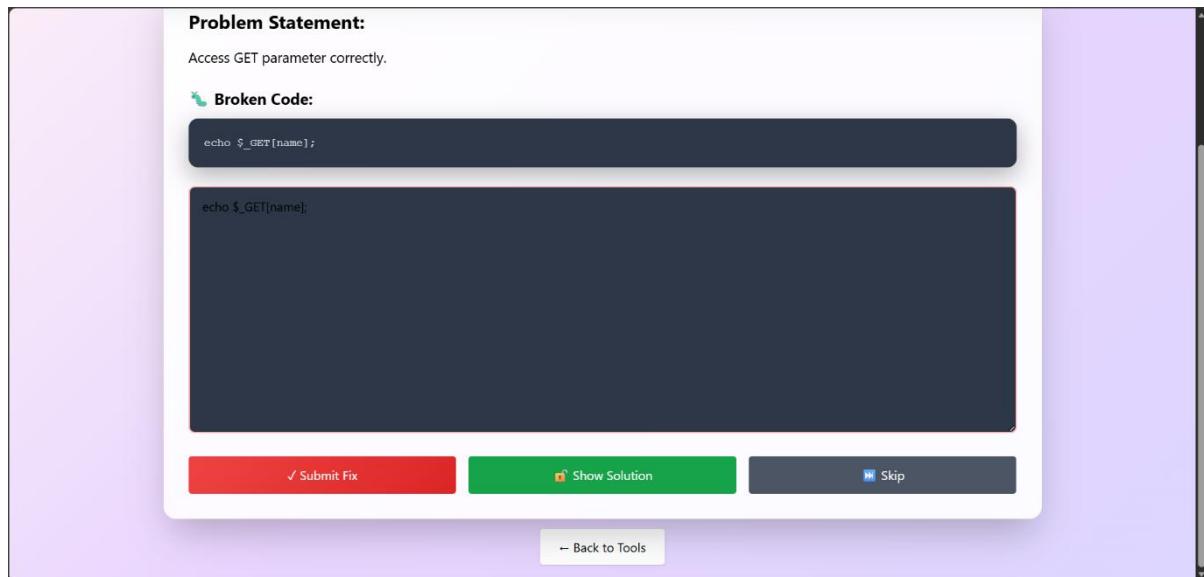


Figure 5.7.27 Fix the Code Page 2

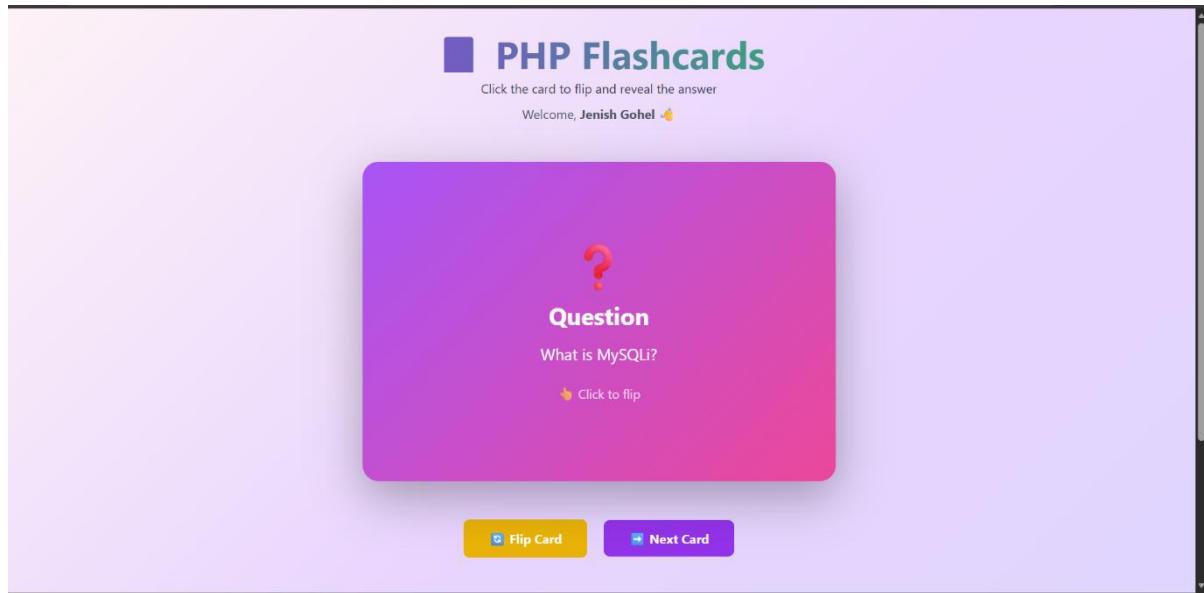


Figure 5.7.28 PHP Flashcards Page

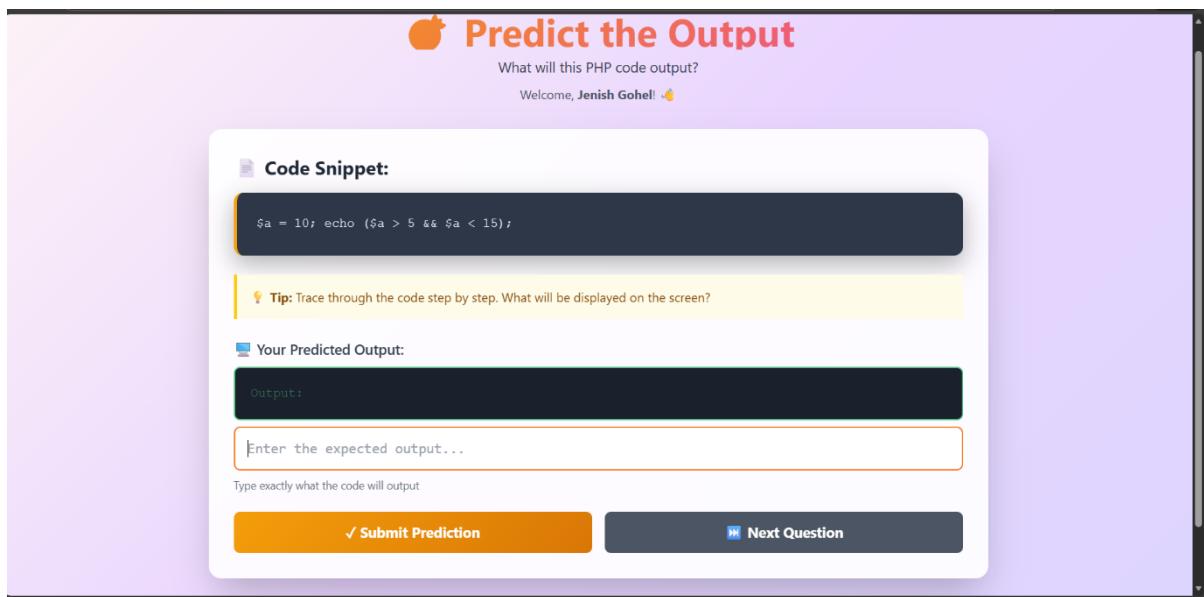


Figure 5.7.29 Predict the Output Page

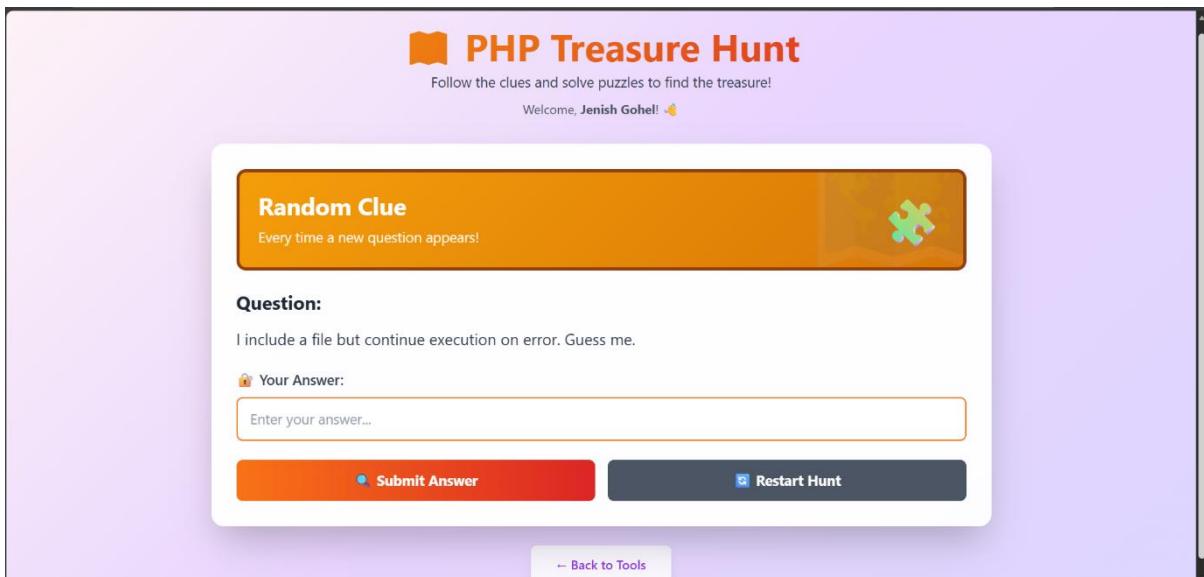


Figure 5.7.30 PHP Treasure Hunt Page

The screenshot shows a web page titled "CHAPTER 1 Introduction to PHP". At the top, there is a navigation bar with links for Home, Tools, Modules (which is the active tab), Feedback, Contact, and Logout. Below the navigation bar, the main content area has a light gray background. It features a heading "CHAPTER 1" and a section title "Introduction to PHP". Underneath the title, there is a sub-section titled "What is PHP?" followed by a brief description: "PHP (Hypertext Preprocessor) is a powerful server-side scripting language designed for web development. It was created by Rasmus Lerdorf in 1994 and has become one of the most popular languages for building dynamic websites." Below this, there is another sub-section titled "Why Learn PHP?" with a bulleted list of reasons: "Widely Used", "Easy to Learn", "Great Career", and "Versatile". Finally, there is a section titled "How PHP Works" with a brief description: "PHP runs on the server side. When a user requests a PHP page: The server processes the PHP code, Generates HTML, Sends the result to the browser".

Figure 5.7.31 Modules Page 1

The screenshot shows a web page titled "How PHP Works". At the top, there is a navigation bar with links for Home, Tools, Modules (which is the active tab), Feedback, Contact, and Logout. Below the navigation bar, the main content area has a light gray background. It features a sub-section titled "How PHP Works" with a brief description: "PHP runs on the server side. When a user requests a PHP page: The server processes the PHP code, Generates HTML, Sends the result to the browser". Below this, there is a code block titled "Your First PHP Code:" containing the following code: <?php echo "Hello, World!"; ?>. At the bottom of the page, there is a footer with the text "Page 1 of 12" and "Welcome, Jenish Gohel!". There are also navigation buttons for "Previous Chapter" and "Next Chapter".

Figure 5.7.32 Modules Page 2

The screenshot shows a web application interface for a learning platform. At the top, there is a navigation bar with icons for Home, Tools, Modules, Feedback, Contact, and Logout. Below the navigation bar, the title "Modules" is displayed. On the left, there is a sidebar with a "Table of Contents" section containing 12 chapters. Chapter 1 is highlighted with a pink background and contains the text "Introduction to PHP". The other chapters are listed as follows:

Chapter	Title
1	Introduction to PHP
2	PHP Basics & Syntax
3	Variables & Data Types
4	Operators
5	Control Structures
6	Loops
7	Arrays
8	Functions
9	Superglobals & Forms
10	Working with Files
11	MySQL & Databases
12	OOP - Object Oriented PHP

At the bottom of the sidebar, there is a button labeled "← Back to Dashboard".

Figure 5.7.33 Modules Page 3

The screenshot shows a feedback form titled "We Value Your Feedback!". The form includes fields for "Your Name" (with placeholder "Guest Learner") and "Your Email" (with a blank input field). There is also a section for rating experience with five yellow stars and a note "Click on stars to rate (1-5)". A pink star icon is visible on the right side of the form. At the bottom, there is a dropdown menu for "Feedback Category" with the placeholder "Select a category...".

Figure 5.7.34 Feedback Page 1

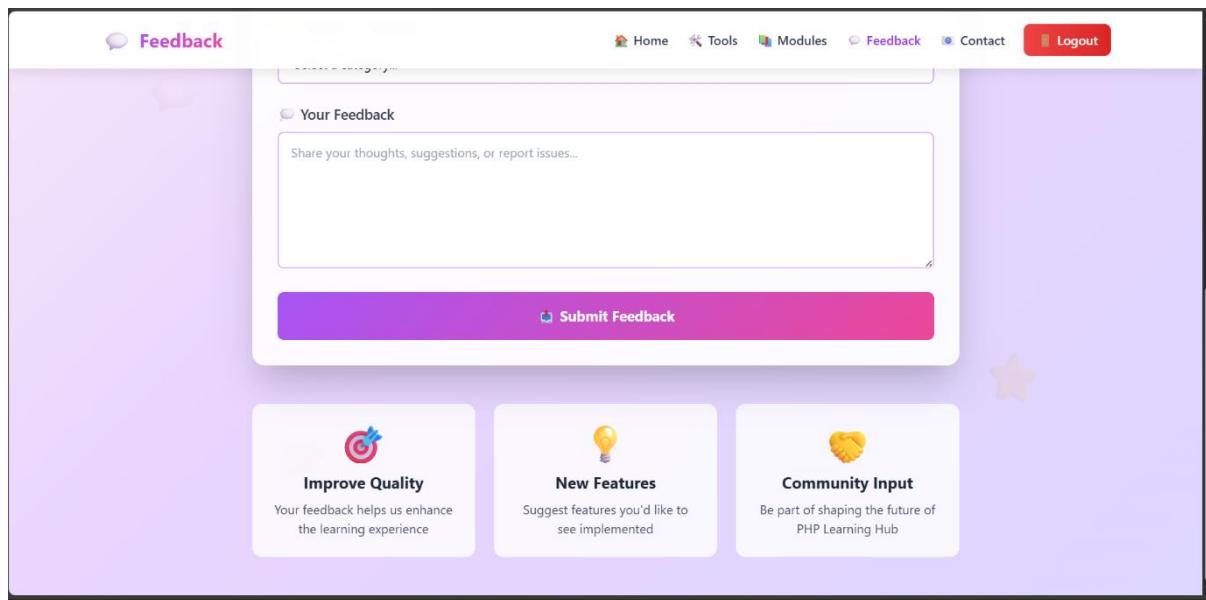


Figure 5.7.35 Feedback Page 2

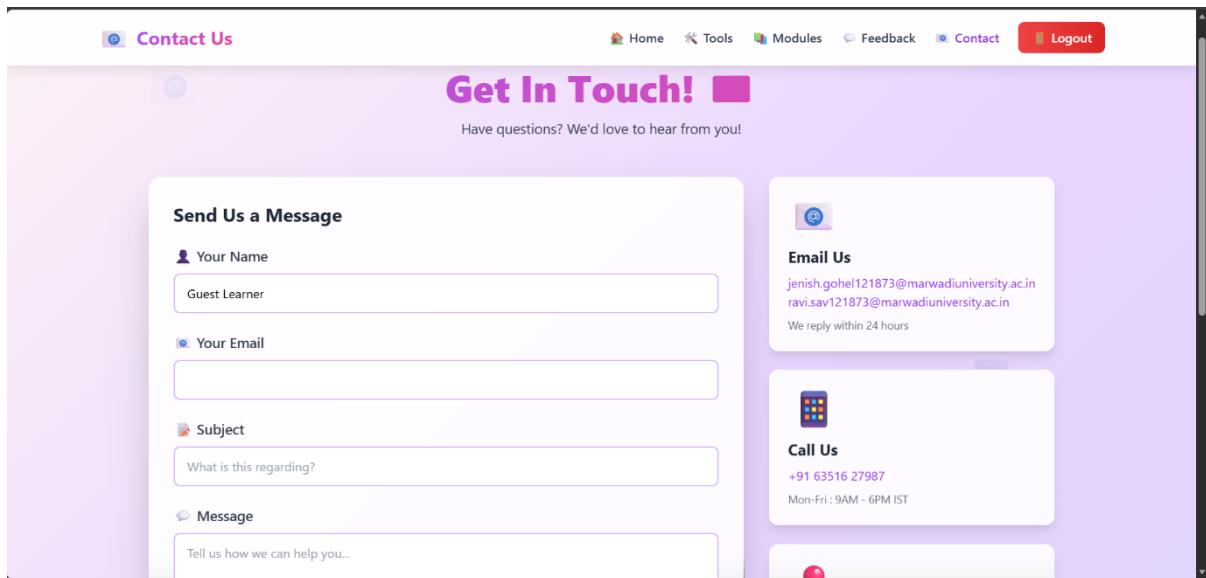


Figure 5.7.36 Contact Page 1

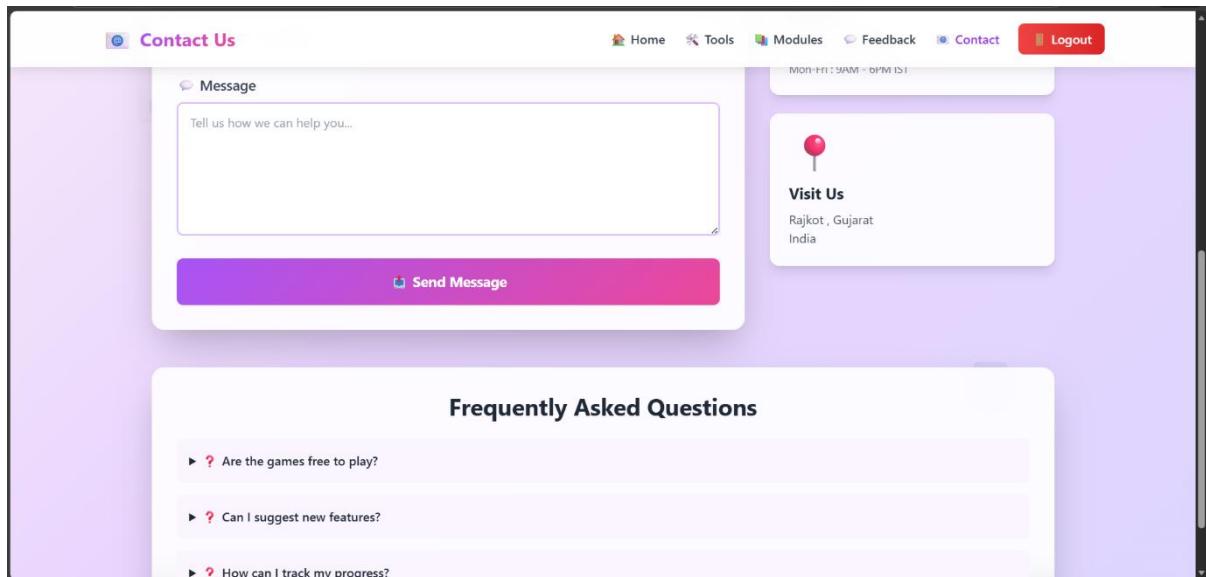


Figure 5.7.37 Contact Page 2

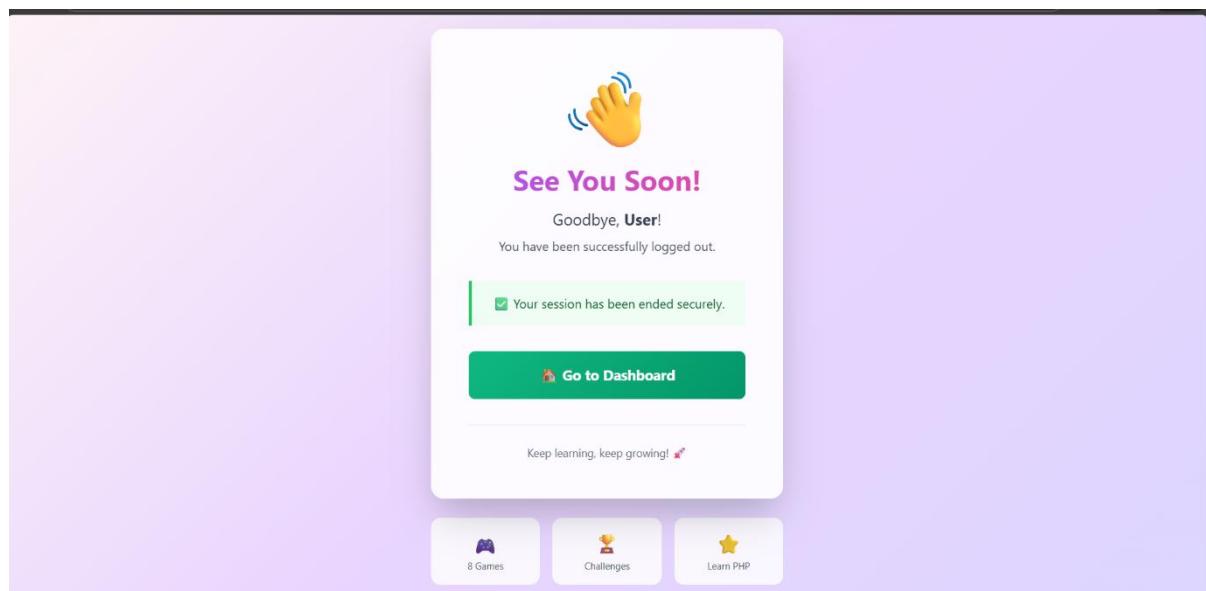


Figure 5.7.38 User Post Logout Page

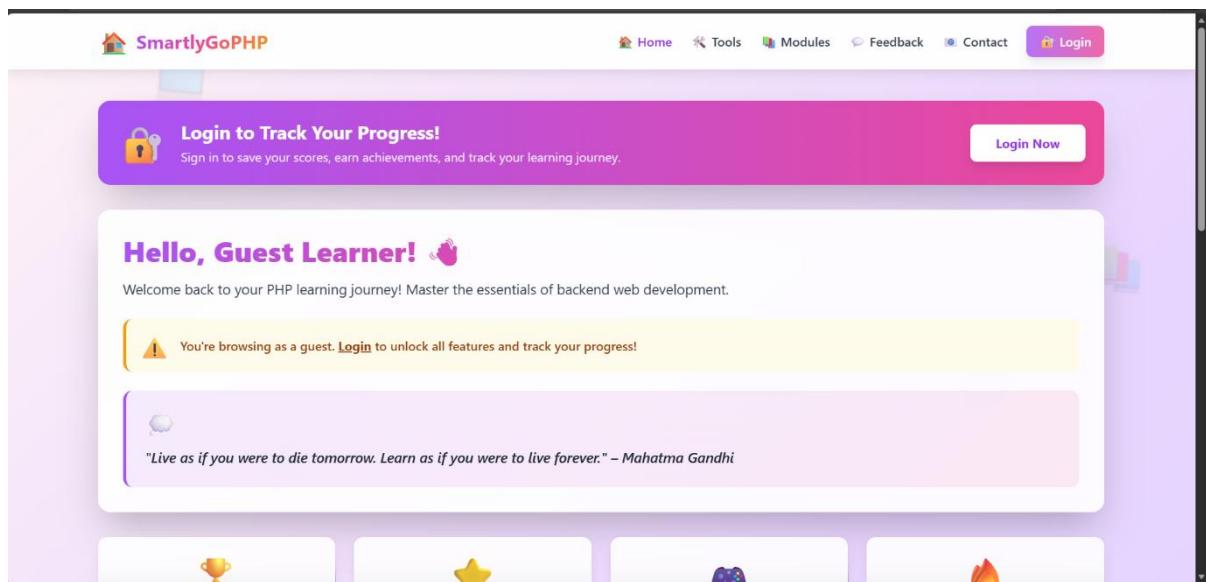


Figure 5.7.39 User without Login Page

6. CONCLUSION

The **SmartlyGoPHP** project successfully demonstrates the development of an interactive and gamified learning platform for understanding **Core PHP concepts**. By combining structured learning modules with engaging tools and games, the system bridges the gap between theoretical knowledge and practical application.

The project provides a well-organized **Admin module** for effective system monitoring and user management, along with a feature-rich **User module** that supports learning, practice, performance tracking, and feedback. The inclusion of score-based challenges, practice tools, progress tracking, and streak management motivates users to learn consistently and improves overall engagement.

SmartlyGoPHP is developed using **Core PHP and MySQL**, ensuring simplicity, reliability, and ease of maintenance. The system's modular design and normalized database structure make it scalable and adaptable for future enhancements. Overall, the project serves as a practical, user-friendly, and educational platform that can be effectively used by students, beginners, and academic institutions to strengthen PHP programming fundamentals.

7. LEARNING DURING PROJECT

The development of the **SmartlyGoPHP** project provided valuable academic and practical learning experiences. The key learnings gained during the project are structured below:

1. Technical Skills Acquired :

- Gained in-depth knowledge of **Core PHP**, including session management, authentication, and form handling.
- Learned secure password storage using hashing techniques.
- Improved understanding of server-side logic and request handling.

2. Database Management Learning :

- Designed and implemented a **relational database** using MySQL.
- Learned about table normalization, primary keys, foreign keys, and indexing.
- Understood real-time data storage for user progress, scores, and streaks.

3. Front-End Development Experience :

- Enhanced skills in **HTML, CSS, and JavaScript**.
- Designed interactive and user-friendly web interfaces.
- Learned to integrate backend data with frontend displays.

4. System Design & Architecture :

- Understood modular system design by separating admin and user functionalities.
- Learned client-server architecture implementation in web applications.
- Improved ability to plan application flow and data interaction.

5. Gamification & Logic Building :

- Implemented score-based and non-score-based learning tools.
- Learned how gamification increases user engagement and motivation.
- Strengthened logical thinking through quiz and coding tool development.

6. Problem-Solving & Debugging :

- Gained experience in identifying and fixing coding and logical errors.
- Improved debugging skills during development and testing phases.

7. Professional & Soft Skills :

- Improved documentation and technical writing skills.
- Learned time management and systematic project execution.
- Gained confidence in presenting and explaining a complete software project.

8. BIBLIOGRAPHY

Online References:

- [1] PHP Group, *PHP Manual*. [Online]. Available: <https://www.php.net/>. [Accessed: Sept. 10, 2026].
- [2] Oracle Corporation, *MySQL Official Documentation*. [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed: Sept. 10, 2026].
- [3] W3Schools, *PHP and MySQL Tutorials*. [Online]. Available: <https://www.w3schools.com/>. [Accessed: Sept. 10, 2026].
- [4] Mozilla Developer Network (MDN), *Web Development Documentation*. [Online]. Available: <https://developer.mozilla.org/>. [Accessed: Sept. 10, 2026].
- [5] Stack Overflow, *Community-Based Troubleshooting Platform for Developers*. [Online]. Available: <https://stackoverflow.com/>. [Accessed: Sept. 10, 2026].

Offline References:

- [6] R. Nixon, *Learning PHP, MySQL & JavaScript*, 5th ed. Sebastopol, CA: O'Reilly Media, 2018.
- [7] Learning during academic laboratory sessions and theory classes, Marwadi University, India.
- [8] PHP & MySQL study materials provided by the University.
- [9] Project development and implementation based on core software engineering principles learned during coursework.
- [10] Conceptual inspiration and learning support from various educational platforms and coding challenge resources.