International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Online course management system

Vaibhav Aggarwal, Laxmi, Yashraj Sharma
Under the Guidance of:
Ms. Shikha Sharma
S.D.College of Engineering and Technology.

Abstract- — The rapid advancement of information technology has transformed the traditional education system into a more flexible and accessible digital learning environment. This research paper presents the design and implementation of an Online Course Management System (OCMS) developed using Django web framework with Python as the backend programming language and SQLite as the database management system. The proposed system provides a comprehensive platform for educational institutions to manage courses, track student progress, handle enrollments, and generate completion certificates automatically. The system implements a role-based access control mechanism supporting three distinct user roles: Administrator, Instructor, and Student. Each role has specific permissions and functionalities tailored to their requirements. The frontend is developed using HTML5, CSS3, and Bootstrap 5 framework, ensuring responsive design across various devices. The research demonstrates how modern web technologies can be leveraged to create an efficient, scalable, and user-friendly learning management system.

Keywords: Learning Management System, Django, Python, Web Application, E-Learning, Course Management, Online Education, SQLite

I. CHAPTER 1: INTRODUCTION

1.1 Background

The education sector has witnessed a paradigm shift from traditional classroom-based learning to digital and online learning platforms. The COVID-19 pandemic further accelerated this transformation, making online education not just an option but a necessity. Learning Management Systems (LMS) have become integral tools for educational institutions, corporate training, and individual skill development.

According to Research and Markets, the global e-learning market size was valued at USD 250 billion in 2020 and is projected to reach USD 1 trillion by 2027, growing at a CAGR of 21%. This exponential growth indicates the increasing acceptance and adoption of online learning platforms worldwide.

An Online Course Management System serves as a centralized platform where educators can create, manage, and deliver course content, while learners can access materials, track their progress, and obtain certifications upon completion. Such systems bridge the geographical gap between teachers and students, providing flexibility in terms of time and location.

1.2 Problem Statement

Traditional educational systems face several challenges:

1. Limited Accessibility: Students in remote areas often lack access to quality education due to geographical constraints.
2. Rigid Scheduling: Traditional classroom-based learning requires students to be physically present at specific times, which may not be feasible for working professionals.
3. Lack of Progress Tracking: Manual tracking of student progress is time-consuming and prone to errors.
4. Resource Management: Managing course materials, assignments, and student records manually is inefficient and challenging.
5. Scalability Issues: Traditional systems cannot easily accommodate a large number of students without significant infrastructure investment.
6. Certificate Management: Manual generation and verification of completion certificates is tedious and susceptible to fraud.

1.3 Objectives

The primary objectives of developing this Online Course Management System are:

1. To design and develop a web-based platform for managing online courses efficiently.
2. To implement role-based access control with three user types: Admin, Instructor, and Student.
3. To provide instructors with tools to create, update, and manage course content including lessons and multimedia materials.

4. To enable students to browse, enroll in courses, and track their learning progress.
5. To implement an automated certificate generation system for course completion.
6. To create a responsive and user-friendly interface accessible across various devices.
7. To ensure data security through proper authentication and authorization mechanisms.

1.4 Scope of the Project

The scope of this Online Course Management System includes:
Included Features:

- User registration and authentication
- Role-based access control (Admin, Instructor, Student)
- Course creation, editing, and deletion (CRUD operations)
- Lesson management within courses
- Student enrollment and unenrollment
- Progress tracking for enrolled courses
- Automated PDF certificate generation
- Responsive web design
- Admin panel for system management
- Excluded Features (Future Scope):
- Live video conferencing
- Real-time chat functionality
- Payment gateway integration
- Mobile application
- AI-based course recommendations

II. CHAPTER 2: LITERATURE REVIEW

2.1 Evolution of E-Learning Systems

The concept of e-learning has evolved significantly over the past few decades:

First Generation (1990s): Computer-Based Training (CBT) systems emerged, primarily using CD-ROMs to deliver educational content. These systems were standalone and lacked interactivity.

Second Generation (2000s): Web-Based Training (WBT) systems became popular with the advent of the internet. Systems like Blackboard and WebCT were introduced, offering basic course management features.

Third Generation (2010s): Cloud-based Learning Management Systems emerged, offering features like video streaming, mobile learning, and social learning. Platforms like Coursera, Udemy, and edX revolutionized online education.

Fourth Generation (2020s): AI-powered adaptive learning systems, virtual reality classrooms, and microlearning platforms represent the current state of e-learning technology.

2.2 Existing Learning Management Systems

Several Learning Management Systems are currently available in the market:

1. Moodle
- Open-source LMS with highly customizable architecture
- Large community support and extensive plugin library
- Requires technical expertise for setup and maintenance

2. Canvas
- Cloud-based LMS with a user-friendly interface
- Strong mobile support and modern design
- Premium pricing for advanced features

3. Blackboard
- Enterprise-level LMS with a comprehensive feature set
- High cost of ownership and complex administration

4. Google Classroom
- Free for educational institutions with simple, intuitive interface
- Limited customization; tightly integrated with Google ecosystem

5. Coursera / Udemy
- MOOC platforms offering wide course variety and professional certificates
- Limited institutional control over course structure

2.3 Comparative Analysis

The table below compares existing systems with the proposed OCMS:

Feature	Moodle	Canvas	Blackboard	Goo. Class.	Proposed
Open Source	Yes	No	No	No	Yes
Cost	Free	High	Very High	Free	Free
Ease of Setup	Medium	Easy	Complex	Easy	Easy
Customization	High	Medium	Medium	Low	High
Role Mgmt	Yes	Yes	Yes	Limited	Yes

Certificate	Plugin	Yes	Yes	No	Yes
Self-Hosted	Yes	No	No	No	Yes

2.4 Research Gap

After analyzing existing systems, the following gaps were identified:

1. Complexity: Most existing LMS platforms are feature-heavy and difficult for small institutions or individual instructors to adopt.
2. Cost: Commercial LMS solutions have high licensing and maintenance costs.
3. Customization: Cloud-based solutions offer limited customization options.
4. Technical Requirements: Open-source solutions like Moodle require significant technical expertise for installation and maintenance.
5. Localization: Many systems lack proper support for regional languages and contexts.

This research addresses these gaps by developing a lightweight, easy-to-deploy, customizable, and cost-effective Online Course Management System using Python and Django framework.

III. CHAPTER 3: SYSTEM ANALYSIS

3.1 Existing System — Disadvantages

- Paper-based records are prone to damage and loss
- Difficulty in tracking student progress manually
- Time-consuming enrollment processes
- No centralized access to course materials
- Manual certificate generation is tedious
- Limited accessibility for remote learners
- Difficulty in managing multiple courses simultaneously
- No real-time updates on course changes

3.2 Proposed System — Advantages

- Centralized digital platform for all course-related activities
- Automated progress tracking with visual indicators
- One-click enrollment and unenrollment
- 24/7 access to course materials from anywhere
- Automated PDF certificate generation
- Role-based access ensuring data security
- Easy course and lesson management for instructors
- Responsive design for mobile accessibility
- SQLite database for easy deployment and backup
- Open-source and fully customizable

3.3 Feasibility Study

3.3.1 Technical Feasibility

The system is technically feasible as it uses Python, Django, SQLite, and Bootstrap — all of which are open-source, well-documented, and widely supported.

3.3.2 Economic Feasibility

All development tools and technologies are free and open-source. The system can be deployed on low-cost hosting with minimal hardware requirements and zero licensing costs.

3.3.3 Operational Feasibility

The user-friendly interface requires minimal training. Clear role separation reduces confusion, and the system can be operated by non-technical staff.

3.4 Requirement Analysis

3.4.1 Functional Requirements

User Management Module:

- FR1: System shall allow users to register with email and password
- FR2: System shall authenticate users during login
- FR3: System shall support three user roles: Admin, Instructor, Student
- FR4: System shall allow users to update their profile information
- FR5: System shall display role-specific dashboard after login
  - Course & Lesson Management:
- FR6–FR10: Instructors can create, edit, delete courses and filter by category
- FR11–FR15: Lessons support text content and video URLs with maintained order
  - Enrollment, Progress & Certificate:
- FR16–FR19: Students can enroll/unenroll; enrollment tracked in dashboard
- FR20–FR22: Lesson completion tracked; progress percentage auto-calculated
- FR23–FR25: PDF certificates generated with student name, course, and unique ID

3.4.2 Non-Functional Requirements

- NFR1: System shall respond to user actions within 3 seconds
- NFR2: System shall support at least 100 concurrent users
- NFR3: System shall be accessible on all modern web browsers
- NFR4: System shall be responsive on mobile devices

- NFR5: User passwords shall be encrypted using secure hashing
- NFR6: System shall maintain 99% uptime

IV. CHAPTER 4: SYSTEM DESIGN

4.1 System Architecture

The system follows the Model-View-Template (MVT) architectural pattern, which is Django's implementation of the Model-View-Controller (MVC) pattern. The architecture consists of six key layers:

- Client Layer (Web Browser): Sends HTTP requests and renders HTML responses
- URL Dispatcher: Maps incoming URLs to appropriate view functions
- Views: Contains business logic, processes requests, and renders templates
- Templates: HTML files with Django template language for dynamic data display
- Models: Defines data structure and handles ORM for database operations
- Database (SQLite): Stores all application data and manages persistence

4.2 Data Flow

At the highest level (Level 0), all user types — Students, Instructors, and the Administrator — interact with the central OCMS, which in turn reads from and writes to the database. At Level 1, four sub-processes handle the flow: User Management, Course Management, Enrollment Management, and Certificate Generation, each communicating with the database for persistence.

4.3 Entity Relationship Diagram

The data model consists of five core entities and their relationships:

- USER (1) → (*) COURSE: One instructor can create many courses
- COURSE (1) → (*) LESSON: One course can have many ordered lessons
- USER (1) → (*) ENROLLMENT: One user can enroll in many courses
- COURSE (1) → (*) ENROLLMENT: One course can have many enrollments
- ENROLLMENT (1) → (*) PROGRESS: One enrollment tracks progress of many lessons

4.4 Database Schema

The system uses five primary database tables: accounts_user, courses_course, courses_lesson, courses_enrollment, and courses_lessonprogress. Key fields include role-based control in the user table, slugified URLs for SEO-friendly course links, ordered lessons within courses, and certificate ID tracking for verified completion.

4.5 Use Case Overview

The Administrator can manage all users, courses, and system configuration. Instructors can create, edit, and delete their own courses and add lessons. Students can browse courses, enroll, track their progress through lessons, and download completion certificates.

V. CHAPTER 5: IMPLEMENTATION

5.1 Technologies Used

Backend

- Python 3.x: High-level, interpreted language with extensive standard library
- Django 4.x: MVT framework with built-in admin, ORM, CSRF, and XSS protection
- SQLite3: Lightweight, file-based ACID-compliant database requiring zero configuration
- ReportLab: Python library for PDF generation used to create completion certificates

- **Frontend**

- HTML5 & CSS3: Semantic markup with Flexbox/Grid layouts and media queries
- Bootstrap 5: Responsive grid system with pre-built components (no jQuery dependency)
- JavaScript: DOM manipulation, form validation, and interactive UI features

- **Development Tools**

- Visual Studio Code: Lightweight editor with Python extension and integrated terminal
- Git: Version control system for tracking code changes
- pip: Python package installer for dependency management

5.2 Module Description

User Authentication Module (accounts app)

Handles all user-related functionalities including custom User model with email as the username field, three roles (admin/instructor/student), profile management, and secure password hashing.

Course Management Module (courses app)
Manages courses, lessons, enrollments, and certificates. Includes CRUD operations for courses with category classification, lesson ordering, slug-based SEO-friendly URLs, and thumbnail support.

Progress Tracking & Certificate Module
Tracks lesson completion with automatic progress percentage calculation. Upon 100% completion, generates a downloadable PDF certificate with a unique ID, student name, course name, and completion date.

5.3 Security Features

- Passwords hashed using PBKDF2 algorithm with SHA256
- Django's built-in CSRF middleware — tokens required for all POST requests
- Template auto-escaping enabled to prevent XSS attacks
- Django ORM parameterizes all queries, preventing SQL injection
- Role-based permissions with view-level and object-level authorization checks

VI. CHAPTER 6: TESTING

6.1 Testing Methodology

The system was tested using a combination of Unit Testing (individual functions), Integration Testing (module interactions), System Testing (complete workflows), and User Acceptance Testing with actual users.

6.2 Test Cases & Results

ID	Test Case	Expected Result	Status
TC01	User Registration (valid data)	Account created successfully	PASS
TC02	Registration with existing email	Error: Email already exists	PASS
TC03	Login with valid credentials	Redirected to dashboard	PASS
TC04	Login with invalid credentials	Error: Invalid credentials	PASS
TC05	Instructor creates course	Course displayed in course list	PASS
TC06	Student creates course (unauthorized)	Access denied + error message	PASS
TC07	Student enrolls in course	Enrollment record created	PASS
TC08	Student enrolls in same course twice	Message: Already enrolled	PASS
TC09	Mark lesson as complete	Progress percentage increases	PASS
TC10	Complete all lessons in course	Course marked complete	PASS
TC11	Download certificate after completion	PDF certificate downloaded	PASS
TC12	Download certificate before completion	Error: Course not completed	PASS
TC13	Instructor edits own course	Course updated successfully	PASS
TC14	Edit course by different instructor	Access denied	PASS
TC15	Delete course with enrollments	Course + enrollments deleted	PASS

6.3 Test Results Summary

Total Test Cases: 15 | Passed: 15 | Failed: 0 | Pass Rate: 100%
All critical functionalities were tested and verified to work as expected. The system handles edge cases and error conditions appropriately, including duplicate enrollment prevention,

unauthorized access control, and certificate download validation.

VII. CHAPTER 7: SCREENSHOTS AND USER INTERFACE

The following screenshots demonstrate the key pages and features of the LearnHub Online Course Management System, captured from a live locally running instance of the application.

7.1 Home Page — Hero Banner

The home page greets visitors with a bold hero banner offering quick access to browse courses or register. Below it, key platform statistics are displayed.

Figure 7.1 — LearnHub Home Page with Hero Section

7.2 User Registration Page

A clean, multi-field registration form collects user details including name, email, password, and role selection (Student or Instructor).

Figure 7.2 — User Registration Form (Create Account)
Figure 7.3 — Registration Form (Password Fields and Validation)

7.3 Login Page

A minimal, focused login form with username and password fields, with a clear call-to-action button and a link to the registration page for new users.
Figure 7.4 — Login Page (Welcome Back Screen)

7.4 Student Dashboard

After logging in, students are greeted with a personalized dashboard showing their enrolled courses count, a list of active courses, and a quick Browse Courses button.
Figure 7.5 — Student Dashboard with User Menu

7.5 User Profile

Each user has a profile card displaying their username, email, and role badge. The profile edit page allows users to update their name, phone, bio, and profile picture.
Figure 7.6 — Student Profile Card (Vaibhav_agg)
Figure 7.7 — Profile Edit Page

7.6 Browse Courses Page

The Browse Courses page features a filter sidebar allowing users to search by keyword, filter by category, and select difficulty level (Beginner, Intermediate, Advanced).

Figure 7.8 — Browse Courses Page with Filter Sidebar
Figure 7.9 — Difficulty Level Filter Dropdown

VIII. CHAPTER 8: CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

This research project successfully designed and implemented an Online Course Management System using Python Django framework. The system addresses the key challenges faced by traditional education systems by providing a digital platform for course delivery, management, and certification.

Key achievements of the project:

1. Successfully implemented a role-based access control system supporting three distinct user roles: Administrator, Instructor, and Student.
2. Developed comprehensive course management features including course creation, lesson management, and content organization.
3. Implemented an efficient enrollment system allowing students to easily enroll and unenroll from courses.
4. Created an automated progress tracking mechanism that accurately calculates and displays learning progress.
5. Developed an automated PDF certificate generation system for course completion recognition.
6. Ensured system security through proper authentication, authorization, and data validation mechanisms.
7. Created a responsive and user-friendly interface using Bootstrap 5.

The system demonstrates the effectiveness of Python and Django in building robust web applications for educational purposes. The modular architecture allows for easy maintenance and future enhancements.

8.2 Future Scope

The following enhancements can be implemented in future versions:

1. Live Video Classes: Integration with video conferencing APIs (Zoom, Google Meet) for real-time online classes and webinars.
2. Discussion Forums: Adding discussion boards for each course where students and instructors can interact and share knowledge.
3. Quiz and Assessment Module: Implementing online quizzes and examinations with automatic grading and feedback.

4. Payment Gateway: Integration with payment gateways (Razorpay, Stripe) for paid courses and subscription models.
5. Mobile Application: Developing native mobile applications for iOS and Android platforms.
6. AI-Based Recommendations: Implementing machine learning algorithms to recommend courses based on user interests.
7. Analytics Dashboard: Advanced analytics showing student engagement, completion rates, and learning patterns.
8. Multi-Language Support: Adding support for multiple languages to cater to a global audience.
9. Gamification: Implementing badges, points, and leaderboards to increase student engagement.
10. RESTful API Development: Creating APIs for integration with third-party applications and LTI support.

CHAPTER 9: REFERENCES

Books

- Shaw, B. and Badhwar, S. (2021). Web Development with Django. Packt Publishing.
- Vincent, W.S. (2022). Django for Beginners: Build Websites with Python & Django. WelcomeToCode.
- Percival, H. (2017). Test-Driven Development with Python. O'Reilly Media.
- Greenfeld, D.R. and Greenfeld, A.R. (2020). Two Scoops of Django 3.x. Two Scoops Press.

Online Resources

- Django Documentation. (2024). Django: The Web Framework for Perfectionists with Deadlines. https://docs.djangoproject.com/
- Python Documentation. (2024). Python 3.x Documentation. https://docs.python.org/3/
- Bootstrap Documentation. (2024). Bootstrap 5 Documentation. https://getbootstrap.com/docs/5.0/
- SQLite Documentation. (2024). SQLite Documentation. https://www.sqlite.org/docs.html
- ReportLab Documentation. (2024). ReportLab User Guide. https://www.reportlab.com/

Research Papers

- Al-Fraihat, D., Joy, M. and Sinclair, J. (2020). Evaluating E-Learning Systems Success. Computers in Human Behavior, 102, pp.67-86.
- Panigrahi, R., Srivastava, P.R. and Sharma, D. (2018). Online Learning: Adoption, Continuance, and Learning Outcome. International Journal of Information Management, 43, pp.1-14.
- Aparicio, M., Bacao, F. and Oliveira, T. (2016). An e-Learning Theoretical Framework. Journal of Educational Technology & Society, 19(1), pp.292-307.
- Martin, F. and Bolliger, D.U. (2018). Engagement Matters: Student Perceptions on Engagement Strategies in Online Learning. Online Learning, 22(1), pp.205-222.

Websites

- Research and Markets. (2021). E-Learning Market - Global Outlook and Forecast 2021-2026. https://www.researchandmarkets.com/
- MDN Web Docs. (2024). HTML, CSS, and JavaScript Documentation. https://developer.mozilla.org/