Construction and Application of Financial Management Network Teaching System in Colleges and Universities under Python Language

Na Wu

874197657@qq.com

Chongqing College of Architecture and Technology, Chongqing 401331, China

Abstract: Online education platforms are integral to educational reforms. This web-based teaching system embodies online education functionalities and process management. Using front-end Vue/React and back-end Python with Django, it includes user and course management modules. ORM technology accesses databases, and features like rich text editors and PDF reporting tools facilitate tasks like teaching resource editing and interactive assignments. Deployed in a university's platform, it enables unified login, resource sharing, and discussions, yielding effective educational outcomes. The system encompasses the full online education process, showcasing technical innovation and practicality.

Keywords: Online education platform, web-based teaching system, front-end and back-end separation, responsive pages

1 Introduction

The development and utilization of online education platforms are crucial in today's educational reforms. Previous research in this field has primarily concentrated on system functionality design. However, a significant gap exists in leveraging technology to address the comprehensive management of the teaching process. This study employs a front-end and back-end separation architecture to design and create a complete web-based teaching system. This system manages the entire teaching process, from course delivery to assignment tracking, demonstrating technological innovation and practicality. This paper details the system's architecture, functions, and application outcomes.

1.1 Presentation Layer

The presentation layer is primarily implemented using front-end technologies such as HTML, CSS, and JavaScript. It is responsible for creating the user interface, receiving user inputs, and presenting the returned results [1-2].As shown in Figure 1:

(1) The front-end page layout utilizes the Bootstrap framework to achieve responsive pages that support both PC and mobile access.

(2) Vue or React is employed as the front-end MVC framework to organize the page structure, implement data binding, and create interactive dynamic effects.

ICMEIM 2023, September 08-10, Wuhan, People's Republic of China

DOI 10.4108/eai.8-9-2023.2340077

(3) The front-end communicates with the back-end using Axios and passes data in the format of RESTful APIs.

(4) Visualization libraries like Echarts are used to create statistical charts and graphs.

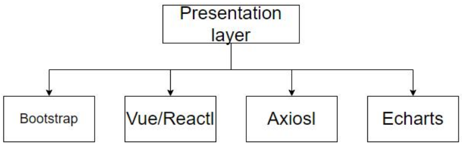

Fig. 1. Presentation Layer

1.2 Business Logic Layer

The business logic layer is implemented using the Python language and the Django framework and is responsible for handling business functionalities and logic.As shown in Figure 2:

User Management Module: Implements functionalities such as user registration, login, session management, and permission control .

(1) Course Management Module: Enables course creation, editing course information, uploading course materials, and assigning assignments [3].

(2) Teaching Resources Module: Stores and distributes various course resource files.

(3) Assignment Management Module: Receives student submissions, allows teachers to grade and provide feedback[4].

(4) Forum Module: Provides an online platform for discussions and exchanges.

(5) Permission Control Module: Implements fine-grained access control based on the RBAC (Role-Based Access Control) model.

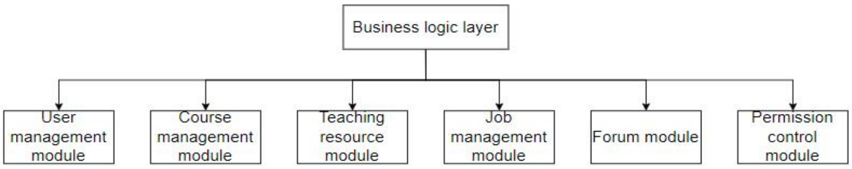

Fig. 2. Business Logic Layer

1.3 Data Access Layer

The data access layer primarily utilizes Django's ORM (Object-Relational Mapping) framework to implement database operations and achieve functionalities for data manipulation [5].As shown in Figure 3:

(1) Connection pooling is employed to enhance database access performance.

(2) It supports both relational database MySQL and non-relational database MongoDB.

(3) Custom data interfaces are encapsulated and provided for the business layer to call[6].

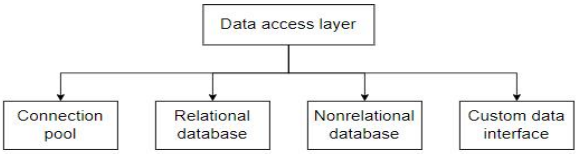

Fig. 3. Data Access Layer

1.4 Database Layer

In the following sections, we delve into the architecture and components of our web-based teaching system. Before we explore the database layer in detail, let's briefly overview the subsequent sections of this paper to provide a clear roadmap for our readers.As shown in Figure 4:

(1) MySQL is used to store relational data such as user information, course details, discussion data, etc.

(2) MongoDB is used to store files, images, and other unstructured data.

(3) Master-slave replication and sharding techniques are employed to enhance data access efficiency.

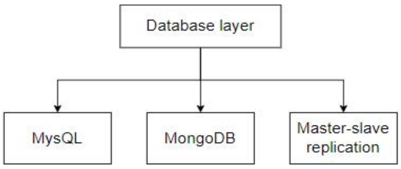

Fig. 4. Database Layer

2 Key Technologies

2.1 Python Web Framework - Django

Django is an efficient and reliable Python web framework widely used for developing database-driven web applications. It implements the MVC (Model-View-Controller) pattern and provides components such as object-relational mapping, template engine, URL routing, and more. This framework significantly reduces repetitive tasks in web development, allowing developers to focus on the application's business logic. Some advantages of using Django for development include a rich ecosystem of third-party components, efficient admin interfaces, reduced database operation code, simplified views and URL handling, and flexible extensibility.Django boasts a strong community support and extensive documentation and has been adopted by many successful websites, making it an ideal choice for building database-driven websites. It can greatly enhance the efficiency and quality of Python web application development.

2.2 ORM (Object-Relational Mapping) Framework - Django ORM

Django ORM is one of the core components of the Django framework, and it implements the mapping between Python objects and the database, greatly simplifying and optimizing database operations in Python web applications. Through ORM, developers can work with the database using an object-oriented approach, using classes and objects to perform CRUD (Create, Read, Update, Delete) operations without the need to write SQL queries. Django ORM provides developers with a convenient database abstraction interface, efficiently mapping models to relational databases, and supporting complex database access operations, including the ability to connect to multiple databases.Using Django ORM can significantly reduce the amount of code required for database operations, lower the complexity of web development, and allow developers to focus on business logic. It seamlessly integrates with Django Admin to generate an admin panel, which can greatly enhance development efficiency. Django ORM is one of the outstanding features of Django, making Python an excellent choice for web application development.

2.3 Rich Text Editor - CKEditor

CKEditor is an excellent web-based rich text editor that offers a WYSIWYG (What You See Is What You Get) editing mode. Users can conveniently create and edit HTML content in the browser, much like using word processing software. CKEditor is known for its user-friendly interface, powerful features, rich plugins, high customizability, and fast loading speed. It also supports comprehensive internationalization. Its open-source license ensures cost-effectiveness, making it a valuable tool for simplifying rich text content management in web applications, enhancing user experiences, and improving content creation efficiency. CKEditor is a top choice for web-based rich text editing and is highly recommended for integration into web applications.

2.4 Data Reporting Tool - ReportLab

ReportLab is a popular Python library for generating reports, offering a robust API to create various document and report formats, including PDF, Excel, Word, and more. It boasts powerful functionality, comprehensive chart support, high customizability, excellent print quality, and ease of integration. With ReportLab, it becomes effortless to transform application data into professional reports, significantly simplifying the development of reporting systems. ReportLab supports multiple languages, provides rich components, and offers strong extensibility to meet complex reporting requirements. It is one of the preferred solutions for Python-based report development, greatly enhancing the value of data analysis and reporting.

3 System Implementation

3.1 System Framework Setup

The framework of this system is built upon the Python language and the Django web framework. The system's backend utilizes Django's database ORM for database access and employs Django's built-in user authentication system to handle functions such as account registration, login, and permission control. For the frontend, the system is developed using HTML/CSS/JavaScript technologies and incorporates jQuery as a JavaScript framework to achieve dynamic interactive effects on web pages. The system integrates CKEditor as a rich text editor, facilitating the publication and editing of text-based teaching content by teachers.During the framework development process, the Model-View-Template (MVT) design pattern is followed, allowing for the decoupling of system functional modules and facilitating future maintenance. Through a well-designed framework, the system achieves role differentiation between teachers and students, organizes teaching content, and provides various teaching modules such as online assignments and tests, thereby offering support for online education. The code is as follows:

(1) Quickly create the project structure using Django's base template.

```
django-admin startproject projectname
```

(2) Configure URL routing to connect with view functions and display basic pages.

```
# urls.py
from django.urls import path
from . import views
urlpatterns = [
    path('', views.index, name='index'),
]
```

Design ORM model classes to establish mappings with database tables.

```
# models.py
class User(models.Model):
    username = models.CharField(max_length=100)
    password = models.CharField(max_length=100)
```

This framework setup establishes the foundation for building the rest of the system's functionalities and features.

3.2 Development of Core Features

The core functionality development of a higher education financial management online teaching system in the Python language can be achieved through a well-structured database design and leveraging Python's rich libraries and frameworks. This primarily includes modules for account management, course management, teaching, assignments, exams, as well as auxiliary functions such as message notifications, data analytics, and permission control.

From a technical perspective, data storage can be handled using database interfaces, while frameworks can be used to organize teaching content and implement user permission control. Modules such as email handling and data analysis can be utilized to implement various functional aspects. It is essential to pay attention to code standards, security, and the creation of well-designed models to build the core functionality of the system.The code is as follows:

(1) Implement user management functionality using class-based views and templates.

```
from django.views import View
class UserView(View):
    def get(self, request):
        return render(request, 'user.html')
```

(2) Utilize the rich text editor to establish the teaching resource addition functionality.

(3) Implement asynchronous interactions using jQuery AJAX.

```
$.ajax({
    url: '/api/course/',
    type: 'GET',
    success: function(data) {
        console.log(data);
    }
});
```

These development steps contribute to building the core functionalities of the online education system, including user management, teaching resource management, and asynchronous interactions.

3.3 Performance Optimization

To optimize the performance of a Python-based higher education financial management online teaching system, various strategies can be employed. These include implementing caching to reduce database queries, optimizing the database with actions like indexing and redundancy, enhancing concurrency using techniques like asynchronous programming and cloud computing, analyzing and optimizing the code for performance bottlenecks, reducing file loading times with CDNs and compression, enabling lazy loading for web pages, and optimizing database connection configurations. By utilizing these Python-based performance optimization methods, the system can deliver a high-performance online teaching experience.

(1) Optimize slow queries using caching mechanisms.

```
from django.core.cache import cache
users = cache.get('users')
if not users:
    users = User.objects.all()
    cache.set('users', users)
```

(2) Enable a database connection pool to reuse database connections.

(3) Load page resources on-demand to accelerate page responsiveness.

```
<script defer src="script.js"></script>
```

These optimization techniques contribute to improving the system's performance by addressing slow queries, efficient database connection management, and enhancing page loading times.

4 Application Example

4.1 System Entrance

The system entrance offers unified login and personalized course display. Through a single portal website, students and teachers can access the system using various login methods. Upon login, users are redirected to the relevant teacher or student portal. After logging in, users see personalized course recommendations and school announcements. Courses are presented as cards with details. Clicking a course card reveals the outline, teacher info, and more. Multimedia-rich announcements can be targeted and timed [7]. The entrance streamlines login, offers tailored course suggestions, and manages announcements for a user-friendly homepage experience.

4.2 Teaching Resources

Teaching resource management enables diverse uploads and cross-device access. Teachers use the backend's rich text editor to upload various formats like PPT, Word, PDF, videos, with breakpoint-resume. Resources have metadata, draft/published states, and chapter-based organization. Published resources are accessible via mobile by scanning QR codes, supporting online viewing and local downloads. Version control is available. This streamlines uploads and ensures consistent resource display, enhancing teaching support.

4.3 Online Assignments

Online assignment management streamlines publishing, completion, and grading. Teachers issue diverse assignments with parameters like questions, instructions, and deadlines, even supporting random question selection [8]. Students edit and submit assignments online, formatting text and attaching files . Teachers can download submissions, give feedback, and grade. Discussions are encouraged among both teachers and students. Assignment statistics provide insights into completion status [9]. This digitizes assignment lifecycles, promoting content sharing and improving teaching efficiency.

4.4 Online Discussions

The online discussion forum promotes focused course communication and feedback. Students and teachers can post anonymously or using real names, utilizing text formatting and multimedia. Discussions are categorized (e.g., assignments, knowledge) and allow interactions. Students freely question and express opinions, while teachers provide prompt @-tagged responses, address trends, and evaluate discussions. Records are saved automatically, encouraging open student-to-student and teacher-to-student interactions [10], fostering interactive and open course teaching.

4.5 Teaching Statistics

The system offers comprehensive teaching data analytics. It tracks student activities, learning curves, and assignment progress using line and pie charts. Student assignment grades are presented in tables. Multi-dimensional statistical analysis includes class averages and standard deviations. Teachers gain insights into student progress and adjust strategies using generated reports. Teaching statistics provide robust data for ongoing evaluations.

5.Conclusion

This study successfully developed a web-based teaching system with a separation architecture. It encompasses core online education functionalities and enhances user experience through features like rich text editors. The system's implementation in a university's platform demonstrated positive teaching outcomes, driven by unified login, course recommendations, resource sharing, interactive assignments, and teaching statistics. The research showcases technological innovation, offering insights for promoting online education platforms and advancing educational reform. Future efforts could refine system architecture, enhance analytics tools, and improve mobile and personalized learning support.

Reference

[1] Yan Q .Design of Teaching Video Resource Management System in Colleges and Universities based on Microtechnology[J].Security and Communication Networks, 2021.

[2] Yu X .The Application of Data Warehouse in Teaching Management in Colleges and Universities[J].IOP Publishing, 2021.

[3] Zhang J .Application of big data collection-analysis-visualization in the teaching process of colleges and universities under the background of the epidemic[J].Journal of Physics: Conference Series, 2021, 1800(1):012009 (7pp).

[4] Hu Y .Research on the Influence of Online Multimedia Corpus Indexing System on English Teaching in Colleges and Universities[C]//CIPAE 2021: 2021 2nd International Conference on Computers, Information Processing and Advanced Education.2021.

[5] Cortes J S .Taming the River: Negotiating the Academic, Financial, and Social Currents in Selective Colleges and Universities (review)[J].The Review of Higher Education, 2012, 35(4):655-656.

[6] Gill T K , Gill S S .Financial management of universities in developing countries[J].Higher Education Policy, 2000, 13(2):125-130..

[7] Huanyang L , Yanlun H , University L .On the Orientation and System Construction of Application-oriented Talents Training of Local Undergraduate Colleges and Universities[J].Educational Research, 2012.

[8] Ulugov B D , Kasimov S U .Application of Pedagogical Information Technologies in the Educational Process of Universities in Uzbekistan[J].International Journal of Information and Communication Technology Education: An Official Pubblication of the Information Resources Management Association, 2021.

[9] Mingdong W .Teaching Quality Management and Evaluation System of Local Universities Under College-enterprise Cooperation[J].Journal of Higher Education, 2018.

[10] Jie Y , University Q .Reflection on the Teaching Governance System in Chinese Colleges and Universities: From the Perspective of Enterprise Multidivisional Structure[J].Review of Higher Education, 2018.