Check for updates

Research on the Teaching Reform of the Website Development Technology Course Based on Knowledge Graph

Ning Zhang and Mingpei Peng*

Guangzhou Huashang Vocational College, Guangzhou 511300, China

*mmmmmikayla@gmail.com

Abstract. In response to the prevalent problems in teaching the core course Website Development Technology for the Software Engineering major, such as fragmented knowledge points, low student engagement, and monotonous teaching methods, this study introduces knowledge graph technology into course instruction. Centered on the core idea of "Knowledge Graph +Website Development Technology+ Project-driven" and taking the project development process as the main thread, a hierarchical knowledge system and a visualized knowledge graph are constructed. On this basis, a three-stage teaching model of "pre-class graph-guided learning — in-class graph-driven instruction — post-class graph-based expansion" is proposed. This model stimulates students' learning interest and initiative throughout the entire pre-class, in-class, and post-class process, significantly enhancing classroom interactivity and instructional precision.

Keywords: Knowledge Graph, Teaching Reform, Software Engineering

1 Introduction

In 1965, Price, constructed a citation network of scientific papers, which is widely recognized as an early prototype of the knowledge graph. The term Knowledge Graph itself was formally popularized by Google in 2012. Knowledge graphs typically store information in graph databases and present it intuitively as graph structures. As a significant branch of artificial intelligence, knowledge graphs integrate multidisciplinary achievements from semantic technologies, data mining, graph theory, and other fields to construct associative networks of cross-domain knowledge[1].

As a core course of the Software Engineering major, Website Development Technology urgently needs to update its teaching content and methods to keep pace with the times. Traditional instruction relies primarily on classroom lectures with abstract content and exhibits notable deficiencies: knowledge points are fragmented and lack systematization, student engagement is low and learning initiative is insufficient, and the teaching method is monotonous, making it difficult to accommodate the personalized learning needs of different students. Moreover, the course involves numerous concepts such as front-end, back-end, and databases, which makes it challenging for students to

Z. Ge et al. (eds.), *Proceedings of the 2026 6th International Conference on Education, Information Management and Service Science (EIMSS 2026)*, Atlantis Highlights in Computer Sciences 25,
https://doi.org/10.2991/978-94-6239-766-8_30

form a systematic knowledge framework during the learning process, thereby directly affecting learning outcomes[2].

As an emerging approach to knowledge representation and organization, knowledge graphs can organize and present knowledge points in the form of graphs, effectively facilitating knowledge integration and comprehension[3]. Centering on the core concept of "Knowledge Graph +Website Development Technology+ Project-driven", knowledge graph technology can be leveraged to systematically construct the course with projects as the driving force, build a course knowledge system, optimize teaching content, enhance teaching interactivity, and thus comprehensively improve teaching quality[4-5].

This paper aims to take project-driven learning as the main thread, integrate knowledge graphs into the Website Development Technology course, transform abstract concepts into visual knowledge structures, help students form clear knowledge threads, achieve dynamic presentation of knowledge points, stimulate learning interest, and increase classroom participation[6-7].This exploration not only taps into students' potential for cognitive development and cultivates their innovation and problem-solving abilities, but also helps promote the transformation and upgrading of traditional teaching models, offering new ideas and approaches for the pedagogical reform of the software engineering discipline.[8-9].

2 Project-Driven Knowledge Graph Construction

2.1 Content Analysis and Knowledge Point Extraction

Ontology design, as a core step in the top-down construction of a knowledge graph, hinges critically on clarifying the hierarchical structure of course knowledge and the associative relationships between concepts. Taking the practical project development process as the main thread, this study systematically conducts ontological modeling of the core knowledge involved in the Website Development Technology course across domains such as front-end, back-end, and databases, delineating five first-level sub-ontologies: Project Fundamentals (encompassing website structure design, development environment configuration, version control, etc.), Front-end Technologies (encompassing HTML5, CSS3, JavaScript, responsive layout, Vue.js, etc.), Back-end Technologies (encompassing Node.js, Express, databases, API design, etc.), Project Integration (encompassing front-end and back-end integration, deployment and launch, project documentation, etc.), and Extension and Optimization (encompassing performance optimization, security protection, SEO, user authentication, etc.). Each first-level sub-ontology can be further refined into second-level and third-level concepts; for example, Front-end Technologies can be decomposed into sub-concepts such as page structure, style presentation, and interaction logic, thereby forming a logically clear knowledge framework, as shown in Table 1.

Table 1. Hierarchical Classification of Knowledge Points for the Website Development Technology Course

First-level Sub-ontology	Second-level Concepts	Third-level Knowledge Points (Examples)
Project Fundamentals	Project planning, environment setup, version management	Website structure design Development environment configuration Version control
Front-end Technologies	Page structure, style presentation, interaction logic, front-end frameworks	HTML5、CSS3 responsive layout、JavaScript、Vue.js
Back-end Technologies	Server-side framework, back-end framework, data persistence, API design	Node.js、Express、Database、API design
Project Integration	System integration, deployment and maintenance, engineering standards, performance improvement	Front-end and back-end integration、Deployment and launch、Project documentation
Extension and Optimization	Performance enhancement, security reinforcement, search engine optimization, user authentication	Performance optimization Security protection User authentication

(In the table, the first-level sub-ontologies are arranged according to the sequential dependencies of the project development process; second-level concepts within the same sub-ontology exhibit parallel or synergistic relationships, while implicit semantic associations such as prerequisite, inclusion, and dependency relationships exist between different sub-ontologies.)

2.2 Knowledge Graph Construction

After completing the hierarchical organization and knowledge point extraction of the Website Development Technology course content, this study employs the py2neo module in Python as a bridge to interact with the Neo4j graph database. Leveraging the flexibility and rich ecosystem of the Python language, a series of graph data operations — including node creation, relationship definition, attribute assignment, and batch import — are performed on the Neo4j database. This process automatically transforms the aforementioned ontology model into a structured knowledge graph, thereby enabling graph-based storage and semantic association of course knowledge.

Taking the Library Book Borrowing Website as an example, this paper illustrates the design of the knowledge matrix (Table 2) and the resulting knowledge graph (Fig. 1), in which the names are abbreviated. With the knowledge graph designed around the practical project development process, students can progressively transition from fundamental theories to technological frontiers, which not only consolidates their professional foundation but also cultivates innovative thinking, aligning with the cognitive patterns of different learning stages and the demands of the industry.

Table 2. Knowledge Matrix of the Library Book Borrowing Website Project

First-level Sub-ontology	Second-level Concepts	Third-level Knowledge Points
Project Fundamentals	Project planning	Requirement analysis, functional module design, database ER diagram design, etc.
	Environment setup	Node.js installation, Git repository initialization, editor configuration

	Version management	Git branch management
Front-end Technologies	Page structure	HTML5 page skeleton
	Style presentation	CSS3 beautification, responsive layout adaptation for mobile devices
	Interaction logic	Form validation, AJAX request encapsulation, DOM manipulation
	Front-end framework	Vue.js component-based development
Back-end Technologies	Server environment	Node.js runtime environment
	Back-end framework	Express route definitions (Book query, Borrow, Return)
	Data persistence	Database design (Book table, User table, Borrowing Record table)
	API design	RESTful API (GET /books, POST /borrow)
Project Integration	System integration	Front-end and back-end API integration, cross-origin handling, integration testing
	Deployment and maintenance	Project packaging, cloud server deployment, process guarding (PM2)
	Engineering standards	API documentation (Swagger), Project README, coding standards
Extension and Optimization	Performance enhancement	Optimization Book list pagination and caching, database index optimization
	Security protection	Password encryption (bcrypt), SQL injection prevention, XSS filtering
	User authentication	User registration/login, JWT authentication, permission control (Admin/Reader)
	User experience	Autocomplete for book search, real-time borrowing status reminders, mobile adaptation

3 Design of the Teaching Model Based on Knowledge Graph

Upon completing the construction of the knowledge graph for the Website Development Technology course, instructors can draw upon the knowledge structure and associative logic embodied in the graph to systematically reorganize teaching resources. Subsequently, throughout the entire teaching process, the associative characteristics and dynamic feedback capabilities of the knowledge graph can be fully leveraged to create a curriculum system that integrates personalization, openness, and precision. During instruction, instructors formulate clear teaching objectives based on the course syllabus, organically integrate the knowledge graph into every phase of classroom teaching, and center on the three stages of "pre-class graph-guided learning — in-class graph-driven instruction — post-class graph-based expansion" to construct an innovative project-driven teaching model architecture grounded in the knowledge graph.

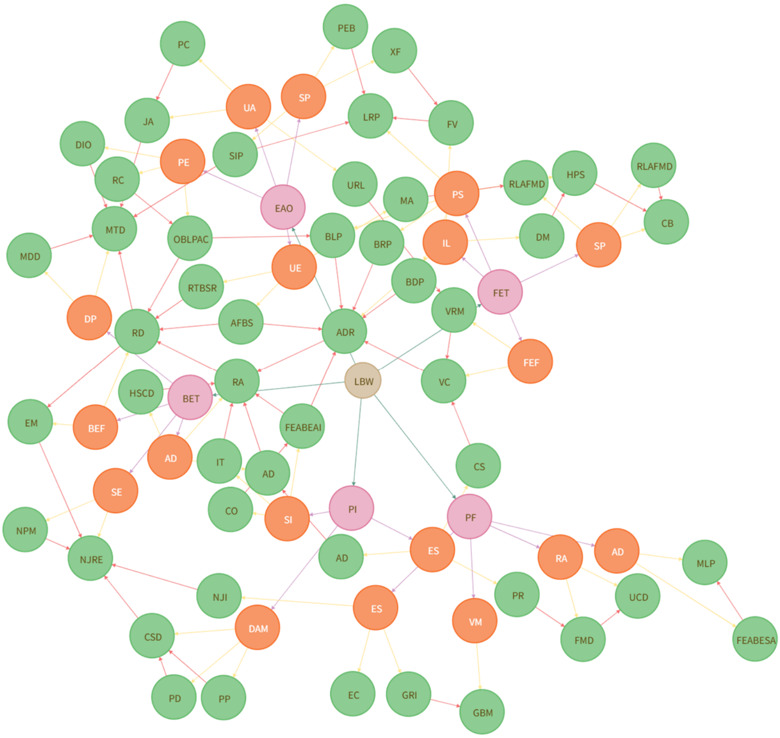

Fig. 1. Knowledge Graph of the Library Book Borrowing Website Project

3.1 Pre-class Graph-guided Learning

During the pre-class stage, drawing upon the course knowledge structure presented by the knowledge graph and the project-driven hierarchical task system, the instructor carefully designs a pre-class learning task list that integrates teaching content, instructional objectives, and discussion topics. This task list is no longer a simple enumeration of isolated knowledge points; instead, it takes Project Fundamentals and other main threads in the graph as the central axis, embedding foundational knowledge such as website structure design, development environment configuration, and version control into project contexts, thereby forming a preview map with inherent logical connections. Based on the task list requirements, students purposefully watch online instructional videos corresponding to the graph nodes and, guided by the knowledge graph, complete study notes and online quizzes. In this process, the knowledge graph provides students with a visualized cognitive scaffold: through the graph, students can intuitively perceive the position of currently learned knowledge points within the entire project development workflow, their pre- and post-requisite relationships, and their connections with subsequent front-end and back-end technologies. Consequently, they not only develop a holistic understanding of the course content but can also accurately pinpoint key learning points and difficulties, thereby initially accomplishing the foundational-level learning tasks within the knowledge system.

3.2 In-class Graph-driven Instruction

During the in-class teaching implementation stage, this study uses the knowledge graph as a structured map, intuitively projecting the complete project-driven development path into a visualized sequence of nodes. It guides students along the technological evolution mainline of "static page construction →interaction logic addition→back-end service integration→database connection→project deployment and launch" ,enabling them to progressively accomplish the core learning tasks of the Website Development Technology course. The design of this pathway is grounded in the hierarchical associations of "Front-end Technologies — Back-end Technologies — Project Integration" within the knowledge graph, transforming the abstract development process into an observable and trackable learning trajectory and helping students maintain a clear awareness of their position and direction within the overall project.

This graph-driven instructional adjustment upgrades traditional experience-based lecturing into dynamic and precise teaching informed by learning behavior data, effectively shortening the students' transformation cycle from confusion to mastery. Meanwhile, instructors can draw upon the difficulty weights of individual nodes in the knowledge graph to timely integrate ideological and political education elements into the curriculum. For instance, they may discuss information security and professional ethics at the User Authentication node, and emphasize engineering standards and responsibility awareness at the Deployment and Launch node, thereby achieving a synergy between knowledge impartation and value guidance[10].

Through the in-class graph-driven project advancement model, the teaching of the Website Development Technology course succeeds in condensing scattered knowledge points into a logical knowledge network that takes the project as its carrier. In the process of learning by doing, students continuously establish deep connections among front-end, back-end, database, and other knowledge domains, while instructors, aided by the knowledge graph, realize a role transformation from teaching knowledge to guiding projects. This substantially enhances instructional precision and students' comprehensive application abilities, laying a solid foundation for subsequent evaluation optimization and competency expansion.

3.3 Post-class Graph-based Expansion

During the post-class expansion and project integration stage, the navigation and scaffolding functions of the knowledge graph extend from the classroom to extracurricular settings, guiding students to deeply internalize knowledge and comprehensively enhance their abilities centered on project objectives. Building on the completion of the main in-class development tasks, and in accordance with the module dependency relationships and functional hierarchies indicated in the knowledge graph, students independently advance the development of the remaining project modules — for example, supplementing expansion modules such as the comment system and user profile center after implementing the core functionalities. This process is not merely a superficial stacking of content, rather, it requires students to draw upon the associated knowledge from multiple nodes in the knowledge graph, such as front-end interaction, back-end data processing, and API design, and to carry out cross-module integration and comprehensive application in authentic project contexts. This deepens their understanding of the holistic nature of the course knowledge system and enables a capability leap from being able to implement individual functions to being able to complete a full project.

Through the graph-based navigation and progressive expansion in the post-class stage, the project-driven teaching model of the Website Development Technology course forms a complete closed loop. Within the positive interaction between independent construction and teacher guidance, students continuously consolidate their knowledge network and strengthen their engineering practice capabilities, thereby laying a solid foundation for subsequent specialized course learning and career development.

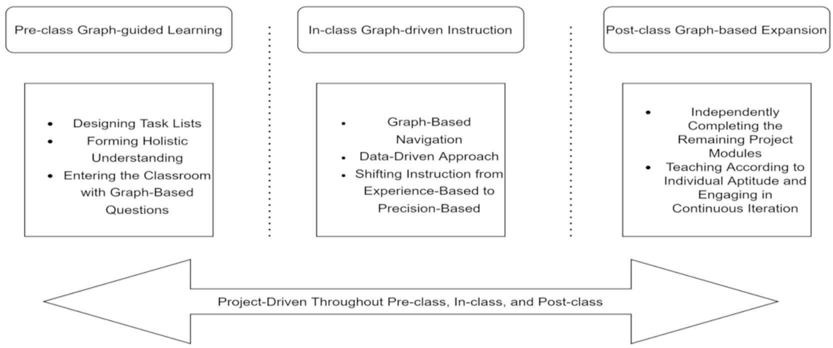

Fig. 2. Innovative Architecture of the Project-Driven Teaching Model Based on Knowledge Graph

Based on the inherent logic of the three stages outlined above, this study constructs an innovative teaching model architecture as shown in Fig. 2. With the course knowledge graph serving as the core support and project-driven learning as the overarching thread, this architecture organically integrates pre-class independent preview, in-class precision teaching, and post-class consolidation and expansion, thereby providing a systematic and actionable practical pathway for the teaching reform of the Website Development Technology course.

3.4 Teaching Applications and Effects

Classroom participation is one of the important indicators for measuring teaching effectiveness[11-12]. In the Website Development Technology course, before the teaching mode reform, instruction was dominated by teacher lectures, and students had limited opportunities for speaking, discussion, and hands-on practice, with relatively uniform modes of participation. Since the implementation of the knowledge graph-based project-driven teaching model, students' classroom participation has significantly increased through components such as pre-class online tasks, in-class problem-driven learning, and group cooperative inquiry.

Statistics on students' classroom performance show that under the traditional teaching model, high-participation students accounted for only 10%, medium-participation students for 55%, and low-participation students for as high as 35%. In contrast, under the innovative teaching model, the proportion of high-participation students surged to 45%, medium-participation students accounted for 40%, and low-participation students decreased markedly to 15%. This set of comparative data strongly indicates that the knowledge graph-based project-driven teaching model can effectively stimulate students' learning interest and awareness of classroom participation, effectively shifting them from passive listening to active participation and deep learning.

The knowledge graph-based project-driven teaching model not only enhances students' classroom participation and academic performance but also promotes the development of comprehensive learning abilities. Students have shown varying degrees of improvement in group collaboration participation and the depth of problem-solving. More importantly, some students with weaker foundations have gradually narrowed the learning gap with their higher-achieving peers under the new teaching model. This outcome fully demonstrates that the teaching model can effectively foster the competency development and academic progress of all students.

4 Conclusion

In the teaching practice of the Website Development Technology course, the deep integration of the knowledge graph and the project-driven teaching model has brought about structural transformation to traditional classroom instruction. By constructing a hierarchical knowledge graph centered on the project development process, instructors can precisely pinpoint the core knowledge points and teaching difficulties of the course, with the key concepts and their intrinsic associations at each level — from Project Fundamentals to Front-end Technologies, Back-end Technologies, Project Integration, and Extension and Optimization — clearly mapped within the graph. For students, the knowledge graph not only serves as a visual cognitive navigation tool that guides them progressively along the development pathway of "static page construction→interaction logic addition→back-end service integration→database connection→project deployment and launch", but also enables them to efficiently and systematically master the core theories and key skills of website development. Through the complete teaching closed loop of "Knowledge Graph + Project-driven" ,students can build a solid engineering foundation for subsequent software engineering courses, lay a firm theoretical and practical groundwork for solving complex software system development problems in the future, effectively enhance their comprehensive analysis ability, engineering practice capability, and innovative exploration spirit, and adapt to the severe challenges and urgent demands posed by the digital-intelligence era for the cultivation of talents in vocational education .

Acknowledgments

This study was funded by the school level AIED project (grant number JWJBGS202522).

References

1. Xiao, Y. (2019). Knowledge graph and cognitive intelligence. Zhangjiang Technology Review, (4), 30-33.
2. Giovana Giardini Borges & Rogéria Cristiane Gratão de Souza.(2024).Skills development for software engineers: Systematic literature review.Information and Software Technology,168,107395-.https://doi.org/10.1016/J.INFSOF.2023.107395.
3. Patrick Rodrigo Da Silva,Érica Ferreira de Souza,Glaúcia Braga e Silva,Giovani Volnei Meinerz & Katia Romero Felizardo.(2025).Applying graph-based knowledge representation to capture insights from discussions forum in software engineering.Science of Computer Programming,244,103298-103298.https://doi.org/10.1016/J.SCICO.2025.103298.
4. Qin, W., & Shen, Y. (2024). Exploration of the curriculum construction path for "Operations Research" based on "knowledge graph + visualized task-driven map". PR World, (7), 157-159.
5. Zhang, M., Yan, Z., Sun, M., Liu, F., & Zhang, X. (2023).Teacher Knowledge Graph: The Inevitable Path of Teachers' Professional Development Empowered by Artificial Intelligence.Modern Educational Technology, 33(8), 38-47.
6. Shen, R., Guo, F., & Wu, J. (2022).Research on the Construction Path ofArtificial Intelligence Empowering Higher Vocational Teaching Resources——FromthePerspective of Connectivism. Education Research Monthly, (7), 52-59. https://doi.org/10.16477/j.cnki.issn1674-2311.2022.07.012.
7. Song, Y., Du, L., Wang, J., & Fan, M. (2026). Research on the Effect Evaluation of Large Model Applications in the Field of Classroom Teaching, 46(2), 44-51. https://doi.org/10.14121/j.cnki.1008-3855.2026.02.005.
8. Zhong, Q. (2022). Instructional Design Based on "Interdisciplinary Accomplishments"-Take STEAM and "Integrated learning" for Instance. Global Education, 51(1), 3-22.
9. Alexander, R. J. (2017). Towards dialogic teaching : rethinking classroom talk (Fifth edition.). Dialogos.
10. Chen, D., Yu, Y., Chen, Z., Wan, X., & Li, H. (2024).Design and Research of Knowledge Graph for Program Design Courses Integrating Ideological and Political Education. Computer & Telecommunication, (7), 62-68. https://doi.org/10.15966/j.cnki.dnydx.2024.07.008.
11. Bhutoria, A. (2022). Personalized education and artificial intelligence in the United States, China, and India: A systematic review using a human-in-the-loop model. Computers and Education: Artificial Intelligence, 3, 100068. https://doi.org/10.1016/j.caeai.2022.100068.
12. Feng, S. (2024). Extension and Symbiosis: A High－quality Development Approaches to Teaching Evaluation System Constructionin Higher Education Institutions in Digital Intelligence Era. Heilongjiang Higher Education Research, 42(2), 84-89. https://doi.org/10.19903/j.cnki.cn23-1074/g.2024.02.007.