Blockchain in Higher Education: a Review on Needed Technologies for an Undergraduate Web3 Course

Nemanja Zdravković[1], Nikola Dimitrijević[1]

[1] *Faculty of Information Technology, Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade, Serbia*

Abstract

Based on our review of the literature and market, we propose an undergraduate course aimed at final-year students in electrical engineering, software engineering, and computer science, with a focus on blockchain technologies as the foundation. This course is designed to provide students not only with theoretical knowledge but also with practical skills necessary for understanding and applying blockchain technologies across various sectors. Our findings indicate that the emphasis on shifting from traditional client-server architectures to decentralized systems, with JavaScript as the leading programming language, is essential for the education of future Web3 engineers and developers. However, we also recognize the importance of including Solidity, the key programming language for the Ethereum blockchain, which enables the development of smart contracts and decentralized applications (DApps). Solidity represents a bridge between traditional programming and the new paradigms brought by blockchain, offering students the opportunity to engage in the development of applications that are not only innovative but also vital for the future of decentralized technologies.

The course will cover fundamental concepts of blockchain, including cryptography, distributed ledgers, consensus algorithms, and blockchain security, as well as advanced topics such as smart contract development, the Ethereum platform, and DApps development using Solidity. Additionally, we plan to integrate case studies and practical projects that will allow students to apply their knowledge in real-world scenarios, preparing them for the challenges and opportunities offered by the job market in the Web3 field. This course will not only provide students with the technical skills needed to work in the blockchain field but will also equip them with critical thinking and an understanding of how blockchain technologies can transform various industries. Our goal is to create a generation of engineers and developers who not only understand the technological aspects of blockchain but can also contribute to its application in innovative and ethically responsible ways.

Keywords

Blockchain, Web3, distributed applications, higher education, Solidity, smart contracts

1. Motivation

Our motivation for this paper comes from the current curricula at Higher Education Institutions (HEIs) aimed at future engineers and developers, which is, in the existing state, lagging in the shift from Web 2.0 to Web3 [1]. Currently, the Internet, comprised of user-generated content, which is the basis of Web 2.0, is experiencing a shift to more decentralized and public environment with blockchain technologies (BCTs) at the forefront [2]. In addition, Web3 will be powered by powerful artificial intelligence (AI) algorithms in order to help empower even more intelligent and adaptive applications [3, 4]. Undergraduate students are still being taught technologies and software architectures which follow Web 2.0 at best, or Web 1.0 at worst, while novel concepts are left for the graduate and PhD level of studies.

Proceedings for the 14th International Conference on e-Learning 2023, September 28-29, 2023, Belgrade, Serbia

EMAIL: nemanja.zdravkovic@metropolitan.ac.rs (A. 1); nikola.dimitrijevic@metropolitan.ac.rs (A. 2);
ORCID: 0000-0002-2631-6308 (A. 1); 0000-0002-8038-0642 (A. 2);

Unfortunately, the current job market for software engineers and developers are in dire need of people who are already proficient in Web3 technologies, even at starting-level positions.

In this paper, we propose an undergraduate level course, aimed at last-year students, which should help close the needed gap between current curricula in Web applications to actual market needs, in accordance with the IEEE/ACM guidelines for Computer Science and Information Technologies Curricula.

2. Methodology

We firstly review relevant literature and guidelines for undergraduate-level curricula in EE/CS/IT, with focus on the technologies with the highest rate of advancement – Web technologies [6].

Afterwards, we compare the data with relevant job positions to cross-reference employers' needs in terms of technologies, i.e. tech-stack.

Finally, we propose the development of two distinct but interconnected curricula to be taught over a single semester at an HEI. The first curriculum is centered on web development and JavaScript, laying the foundational knowledge and skills essential for understanding and engaging with modern web technologies. This curriculum is designed not only to provide comprehensive training in web development but also to serve as a prerequisite for the second curriculum.

The second curriculum delves into blockchain technology (BCT) and Web3, building upon the web development principles acquired in the first course. This advanced curriculum emphasizes the disruptive potential of BCTs in the Web3 landscape. It focuses on decentralization principles, smart-contract programming languages, and the practical applications of these technologies in creating decentralized applications.

Furthermore, given the increasing relevance of edge devices in the Web3 ecosystem, it is crucial for students to gain an understanding of the inner workings of battery-powered end user devices. Knowledge of wireless mobile communications standards and protocols, particularly those pertaining to 5G and beyond, is also integral. This understanding is essential as these technologies form the basis of communication and interaction in the Web3 domain.

Together, these curricula provide a comprehensive educational pathway, equipping students with the necessary skills and knowledge to excel in the evolving field of Web3 and blockchain technologies.

3. Curriculum Design for Web course

This chapter delves into the core technologies of JavaScript (JS) and React, which form the foundation of modern web development, especially in the context of Web3 applications. Over a 15-week course, with 7 weeks dedicated to JavaScript and 8 weeks to React, students will gain a comprehensive understanding of these technologies, their evolution, and their application in the rapidly growing field of Web3 development.

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language that is a core component of the World Wide Web. Initially designed for client-side web development, JavaScript has evolved to become a versatile language used in various environments, from server-side with Node.js to mobile app development [7, 8].

React, developed by Facebook, is an open-source JavaScript library for building user interfaces, particularly for single-page applications. It allows developers to create large web applications that can change data without reloading the page [9, 10].

The combination of JavaScript and React is particularly powerful in the development of decentralized applications (dApps) on blockchain platforms. JavaScript's versatility and React's efficient rendering make them ideal for creating interactive and responsive web interfaces for blockchain-based applications [11].

Table 1
Course structure for students focusing on web development, with 3 weeks dedicated to cryptography, 7 weeks dedicated to JavaScript and and 8 weeks dedicated to React.

Week	Lessons
Week 1: Introduction to JavaScript	Lesson 1: Overview of JavaScript in Web Development Lesson 2: Setting Up the Development Environment Lesson 3: Basic Syntax and Variables
Week 2: Control Structures and Data Types	Lesson 4: Conditional Statements (if, else, switch) Lesson 5: Loops (for, while, do-while) Lesson 6: Understanding Data Types and Operators
Week 3: Functions and Scope	Lesson 7: Writing and Calling Functions Lesson 8: Scope and Closure Concepts Lesson 9: Arrow Functions and this Keyword
Week 4: Objects and Arrays	Lesson 10: Working with Objects Lesson 11: Manipulating Arrays (map, filter, reduce) Lesson 12: JSON and Object-Oriented Programming Concepts
Week 5: Asynchronous JavaScript	Lesson 13: Understanding Asynchronous Programming Lesson 14: Promises and Async/Await Lesson 15: Making HTTP Requests (AJAX, Fetch API)
Week 6: Advanced JavaScript Concepts	Lesson 16: ES6+ Features (Spread/Rest, Destructuring) Lesson 17: Modules and Import/Export Lesson 18: Error Handling and Debugging
Week 7: JavaScript in Web3	Lesson 19: Introduction to Web3 and JavaScript Lesson 20: Connecting JavaScript with Blockchain (Basics) Lesson 21: Overview of Smart Contracts and DApps
Week 8: Introduction to React	Lesson 22: React Fundamentals and JSX Lesson 23: Components and Props Lesson 24: State and Lifecycle Methods
Week 9: React State Management	Lesson 25: Handling State in React Lesson 26: Using Context API Lesson 27: Introduction to Redux
Week 10: React Router and SPA	Lesson 28: Single Page Applications with React Router Lesson 29: Dynamic Routing and Navigation Lesson 30: Managing Route Parameters
Week 11: Advanced React Concepts	Lesson 31: Higher-Order Components (HOCs) Lesson 32: Render Props and Custom Hooks Lesson 33: Utilizing useEffect and Custom Hooks
Week 12: React and API Integration	Lesson 34: Fetching Data from APIs in React

	Lesson 35: Handling API Responses and State Updates Lesson 36: Error Handling in API Calls
Week 13: Building a React Application	Lesson 37: Planning and Structuring a React App Lesson 38: Building a Functional Component Lesson 39: Styling in React (CSS, Styled Components)
Week 14: React in Web3	Lesson 40: Integrating React with Web3 Lesson 41: Building a Front-End for DApps Lesson 42: Interacting with Smart Contracts in React
Week 15: Final Project and Review	Lesson 43: Final Project Overview Lesson 44: Project Development and Implementation Lesson 45: Course Review and Future Learning Pathways

This chapter underscores the importance of JavaScript and React as foundational technologies in Web3 development. Through the 15-week course, students will not only learn the theoretical aspects of these technologies but also apply them in practical projects, preparing them for the challenges and opportunities in the field of Web3.

4. Curriculum Design for blockchain Web3 course

This chapter provides an in-depth exploration of the fundamental technologies and concepts underpinning Web3 development: cryptography, blockchain, smart contracts, and Solidity. These elements are crucial for understanding the infrastructure and development of decentralized applications (dApps) and blockchain systems.

Over a 15-week course, with 3 weeks dedicated to cryptography, 6 weeks dedicated to basics of blockchain technology and Ethereum and 8 weeks to React, students will gain a comprehensive understanding of these technologies, their evolution, and their application in the rapidly growing field of Web3 development.

Cryptography is not just a tool for securing information, but the very foundation upon which the trust and integrity of blockchain technology are built. It ensures the confidentiality, integrity, and authenticity of transactions in a decentralized environment. Cryptography has evolved from simple ciphers used in ancient times to complex algorithms that secure modern digital communications. Its role in blockchain is pivotal, as it encrypts transactions and creates a secure and immutable ledger [12].

Key cryptographic algorithms used in blockchain include hash functions, public-key cryptography, and digital signatures. Hash functions like SHA-256 in Bitcoin ensure data integrity, while public-key cryptography and digital signatures provide secure transaction mechanisms [13].

Blockchain technology is a distributed ledger that records transactions across many computers, ensuring that records cannot be altered retroactively without altering all subsequent blocks. Blockchain's decentralized nature eliminates the need for a central authority, making it inherently resistant to data manipulation. This technology underpins cryptocurrencies like Bitcoin and Ethereum, and its applications extend to various sectors like finance, healthcare, and supply chain [14]. Blockchain technology has evolved from its first-generation blockchain (Bitcoin) to second-generation (Ethereum) with smart contract capabilities, and now to third-generation blockchains focusing on scalability and interoperability [15].

Ethereum is a revolutionary blockchain platform that extends beyond the capabilities of a simple cryptocurrency like Bitcoin. It was proposed in late 2013 by Vitalik Buterin and development was crowdfunded in 2014, leading to its live launch in 2015. Ethereum distinguishes itself through its implementation of smart contracts, which are self-executing contracts with the terms of the agreement directly written into lines of code [16].

At the heart of Ethereum's functionality is the Ethereum Virtual Machine (EVM), which is the runtime environment for smart contracts in Ethereum. It is a powerful, sandboxed virtual stack embedded within each full Ethereum node, responsible for executing contract bytecode. Contracts in Ethereum are written in high-level programming languages, such as Solidity, and are then compiled into bytecode, which the EVM can read and execute [17]. Ethereum also introduces the concept of 'gas', which is a unit that measures the amount of computational effort required to execute operations like transactions or smart contracts. Every operation in Ethereum requires a certain amount of gas, with more complex operations requiring more gas. Gas is paid in Ethereum's native currency, Ether (ETH), and it incentivizes miners to process and validate transactions on the network [18].

Ethereum's ability to support decentralized applications (dApps) through smart contracts has opened up a wide range of possibilities, from decentralized finance (DeFi) to non-fungible tokens (NFTs), and more. This functionality has positioned Ethereum as a leading platform for blockchain-based innovations [19].

Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They run on the blockchain and automatically execute when predetermined conditions are met. Smart contracts enable the automation of complex processes in a transparent and conflict-free environment, reducing the need for intermediaries. They are pivotal in applications like decentralized finance (DeFi) and non-fungible tokens (NFTs) [20].

Solidity is a statically-typed programming language designed for developing smart contracts that run on the Ethereum Virtual Machine (EVM). It is influenced by C++, Python, and JavaScript and is designed to target the unique requirements of blockchain-based applications. Solidity's syntax is similar to that of JavaScript, making it relatively easy for developers with a background in web development to learn. It allows for the creation of complex contracts for decentralized applications, including voting systems, crowdfunding platforms, and automated token exchanges [21].

Table 2

Course structure for students focusing on blockchain and Web3 development, with 3 weeks dedicated to cryptography, 3 weeks dedicated to Basics of Blockchain Technology and Ethereum Systems and eight weeks dedicated to Basics of Smart Contracts.

Module	Lesson
Module 1: Cryptography	Lesson 1: Introduction to Cryptography
Module 1: Cryptography	Early Cryptography and First Algorithms
Module 1: Cryptography	Hash Functions and Practical Exercises
Module 2: Basics of Blockchain Technology and Ethereum	Lesson 4: The History of Money and Cryptocurrency
Module 2: Basics of Blockchain Technology and Ethereum Systems	Lesson 5: Digital Ledger Systems
Module 2: Basics of Blockchain Technology and Ethereum Systems	Lesson 6: Decentralized Consensus
Module 2: Basics of Blockchain Technology and Ethereum Systems	Lesson 7: The Evolution of Blockchain Technology
Module 2: Basics of Blockchain Technology and Ethereum Systems	Lesson 8: Introduction to Ethereum
Module 2: Basics of Blockchain Technology and Ethereum Systems	Lesson 9: Ethereum's Mechanics and Gas Concept
Module 3: Basics of Smart Contracts	Lesson 10: Introduction to Smart Contracts
Module 3: Basics of Smart Contracts	Lesson 11: Working with the Remix Environment

Module 3: Basics of Smart Contracts	Lesson 12: Smart Contract Basics: Data Types, Functions, and Operations
Module 3: Basics of Smart Contracts	Advanced Smart Contract Concepts: Deployment, ABI Interfaces, Test Networks, and Test Ether
Module 3: Basics of Smart Contracts	In-Depth Smart Contract Development: Memory, Storage, Calldata, Constants, Inheritance, and Polymorphism
Module 3: Basics of Smart Contracts	Lesson 15: Interacting with Smart Contracts: ERC20 and ERC721 Standards, Contract Interactions, and Abstract Contracts

5. Conclusion

Our research has highlighted a significant gap in the availability of comprehensive Web3 curricula at the undergraduate level. While many Higher Education Institutions (HEIs) offer advanced courses on these topics at the Master's or PhD levels, there is a clear need for introducing Web3 concepts earlier in the academic journey, specifically in the final years of undergraduate studies. This early introduction is crucial for aligning educational outcomes with the evolving demands of the Web3 job market. Our proposed course framework, with its dual-curriculum approach, addresses this gap effectively. The first curriculum, focusing on web development and JavaScript, lays a solid foundation in the principles of modern web technologies. This foundation is essential not only for understanding the technical aspects of Web3 but also for ensuring students are well-prepared for the more advanced topics covered in the second curriculum.

The second curriculum delves deeper into blockchain technologies, which are increasingly recognized not just for their role in cryptocurrencies but more importantly for their application in decentralized systems - the very cornerstone of Web3. This curriculum is designed to be comprehensive and rigorous, spanning 15 weeks and adhering to the guidelines set by IEEE/ACM for Computer Science and Information Technologies Curricula.

In our curriculum, JavaScript emerges as the primary programming language due to its effectiveness in handling peer-to-peer applications, a key component of decentralized systems. Additionally, we incorporate newer languages that are gaining prominence in the realm of smart contracts, such as Solidity and Rust. These languages are critical for students to master in order to develop secure, efficient, and functional decentralized applications. Furthermore, we emphasize the importance of integrating topics related to security within the curriculum. As Web3 technologies offer enhanced data security, particularly in the handling of personal data, it is imperative for students to understand the security challenges and solutions inherent in these technologies. This knowledge is not only academically relevant but also crucial for preparing students to address real-world security concerns in their future careers in the Web3 domain.

In conclusion, our proposed curricula are designed to bridge the current educational gap in Web3 technologies at the undergraduate level. By providing a structured and comprehensive learning pathway, we aim to equip the next generation of engineers and developers with the skills and knowledge necessary to thrive in the rapidly evolving Web3 job market.

6. Acknowledgements

This paper was supported in part by the Blockchain Technology Laboratory at Belgrade Metropolitan University, Belgrade, Serbia, and in part by the Ministry of Education, Science and Technological Development, Republic of Serbia (Project III44006).

7. References

[1] Chisega-Negrilă, Ana-Maria. "Education in web 3.0." JADLET Journal of Advanced Distributed Learning Technology (2013): 50-59.

[2] Cao, Longbing. "Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci." IEEE Intelligent Systems 37.3 (2022): 6-19.

[3] Voshmgir, Shermin. Token economy: How the Web3 reinvents the internet. Vol. 2. Token Kitchen, 2020.

[4] Filipčić, S. "Web3 & DAOs: an overview of the development and possibilities for the implementation in research and education." 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2022.

[5] Murray, Alex, Dennie Kim, and Jordan Combs. "The promise of a decentralized internet: What is Web3 and how can firms prepare?." Business Horizons (2022).

[6] Kimbrell, Julia. The impacts of web 2.0, web 3.0, and web 4.0 technologies used in distance education. East Carolina University, 2013.

[7] D. Flanagan, JavaScript: The Definitive Guide, O'Reilly Media, 2020.

[8] M. Haverbeke, Eloquent JavaScript: A Modern Introduction to Programming, No Starch Press, 2018.

[9] A. Banks, & E. Porcello, Learning React: Modern Patterns for Developing React Apps, O'Reilly Media, 2020.

[10] M. Nazare, & J. Murray, React and Web3: Building Complete Web3 Apps with React, Independently Published, 2021.

[11] D. Dietrich, Building Ethereum Dapps: Decentralized Applications on the Ethereum Blockchain, Manning Publications, 2021.

[12] J. Katz, & Y. Lindell, Introduction to Modern Cryptography, CRC Press, 2020.

[13] A. J. Menezes, P. C. van Oorschot, & S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[14] A. Narayanan, et al., Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.

[15] W. Mougayar, The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, Wiley, 2016.

[16] Buterin, V. (2014). Ethereum White Paper. Ethereum Foundation.

[17] A. Greco, Mastering Ethereum Virtual Machine, Packt Publishing, 2019.

[18] C. Dannen, Ethereum for Web Developers, Apress, 2017.

[19] W. Mougayar, The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, Wiley, 2016.

[20] C. Dannen, Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners, Apress, 2017.

[21] A. M. Antonopoulos, & G. Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 2018.