From Coursework to Industry: Improving UVA's Full Stack Web Development Offerings

CS 4991 Capstone Report, 2025

Simon Anderson
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
uhx8bu@virginia.edu

ABSTRACT

Modern web development practices and frameworks are critical for preparing computer science students for industry, yet the current UVA curriculum lacks emphasis on these technologies. To address this gap, I propose the implementation of a new full stack web development course that integrates modern tools, frameworks, and practices used in industry. This proposal will build on insights from analyzing and comparing UVA's existing courses in this realm to an external program I participated in that focused on modern web development. The proposed course would combine hands-on experience with popular frameworks like React, Node.js, and Docker, and cover important topics such as RESTful APIs, CI/CD pipelines, and scalable deployment practices. Methodologies in this course would primarily include hands-on project-based learning, working in teams with other developers, and case-studies which look at real-world best practices in action. Anticipated outcomes include a higher degree of student readiness for industry roles which require modern web development skills, as well as a more relevant curriculum which aligns with industry standards. Future work will involve creating course material, getting approval to run the course, piloting the course, and gathering feedback from students and professionals to refine it over time.

1. INTRODUCTION

Graduates of UVA's computer science program enter the job market with a strong foundation in theoretical concepts such as data structures, algorithms, binary arithmetic, state machines, object-oriented programming, and more. However, many find themselves unprepared for the practical demands of modern software engineering roles. Entry level positions often expect familiarity with frameworks like React, RESTful APIs, and containerization tools like Docker. Many of these skills are not emphasized in the current curriculum, and this disconnect leaves many students struggling to bridge the gap between academic knowledge and industry expectations.

While foundational computer science concepts are still important, the lack of formal training in applied software development limits student readiness for these real-world engineering tasks. Many universities, including UVA, continue to prioritize theoretical coursework over hands-on experience with modern tools. Even when they do prioritize these hands-on tasks, it is often with outdated tech stacks, including PHP, old JavaScript syntax and more. As a result, students, including myself, must rely on self-teaching and external programs. I participated in an intensive program called Launch, in which I learned fundamentals of full-stack development over the course of three weeks. This was where I have gained most of my

modern web development knowledge, and without structured guidance like this, the learning process can be inconsistent and inefficient. As a direct result, it can be more difficult for UVA students and graduates to gain any advantage in a competitive job market.

2. RELATED WORKS

Computing education plays a crucial role in preparing students for the evolving job market, yet curricula continue to lag behind. Allison (2023) highlights the understanding that students have access to "fifty-one vocational and technical qualifications in computing" before university, but not all of these qualifications give students the skills employers seek. In fact, an employer skills survey found that nearly "30% of skill-shortage vacancies were attributed to a lack of digital skills," showcasing the gap between education and workforce demands. Institutions also face constraints such as resource limitations, staff expertise and assessment structures, making it difficult to offer courses that fully align with industry needs. Accordingly, it is important to balance real-world learning topics and experiences with the feasibility of expanding curriculum and increasing faculty.

Saurabh (2024), the founder and CEO of Auditzy, emphasizes the importance of a project-based approach in web development education, which "elevates web development education from theoretical to practical" by simulating real-world industry experiences. He highlights how this approach helps learners integrate various skills and develop not just coding proficiency, but also useful soft skills like problem-solving and teamwork. His point about providing "clear guidelines, expectations, and milestones" is crucial for enabling success and aligns with the structure of the proposed course. In all, his philosophy meshes with the guided, hands-on, project-based curriculum I propose.

Lastly, according to a 2021 survey of industry professionals (Stack Overflow, 2021), modern frameworks are becoming increasingly important in web development. React.js, for example, was identified by 41.4% of respondents as the most important framework, surpassing jQuery, which was considered by only 34.52%. This was the first year in which React had been ranked first, and this shift highlights the need for students to be proficient in contemporary technologies. Additionally, cloud platforms play a significant role in modern development, with 54.22% of respondents identifying AWS as an essential platform, followed by Google Cloud and Microsoft Azure at approximately 30% each. This data aligns with the proposed course, which aims to equip students with the skills to use not only modern frameworks but also important cloud platforms and industry best practices.

3. PROPOSAL DESIGN

To address the gap in UVA's computer science curriculum, I propose the development of a comprehensive full-stack development course. This course will provide students with hands-on experience using modern industry-standard tools and frameworks, equipping them with necessary skills for today's job market. The course will be structured into multiple units, each focusing on a critical aspect of full-stack development. Each unit will include a mini-project designed to reinforce the topic of that particular unit, and students will also work on a semester-long group project that integrates these concepts into a fully functional web app.

3.1 Course Structure and Units

The course will begin with an introduction to web development and version control, where students will familiarize themselves with the concepts of full-stack development, Git/GitHub, command-line basics, and a refresher on HTML and CSS. During this unit, students will create a simple personal website

only run locally, while also setting up repositories and beginning to plan with groups for their semester-long project.

Next, students will dive into front-end development with React. They will learn JavaScript ES6 best practices, JSX, component-based architecture, state management, and React hooks. Through this unit, they will develop a small React application, such as a to-do list or weather app, while also designing the frontend architecture and UI components for their group project.

The third unit will introduce backend development with Node.js and Express. Students will learn about creating server-side applications, RESTful APIs, the Express framework, and middleware. They will develop a basic REST API for a book collection as their mini-project, and they will begin coding and integrating the backend for their full-stack application.

Following this, the students will study database management with both SQL and NoSQL methods, focusing on PostgreSQL and MongoDB. They will practice CRUD operations and implement the backend database interactions for a partially completed polling app started by the professor. At the same time, they will be designing and integrating databases into their semester-long project, making sure the app has a sound data structure. At this point, app should be able to perform its core functionalities with data persistence.

Authentication and security will be the focus of the fifth unit. Students will learn about user authentication using JWT and OAuth, hashing passwords, and implementing role-based access control. Their mini-project will involve adding authentication to a simple application and displaying different things in the app based on different user roles. This is also when students will add authentication, security measures, and role management into their group projects.

In the sixth unit, students will learn about deployment, CI/CD pipelines, and cloud services. They will learn how to use Docker, AWS, and Kubernetes. They will also set up continuous integration and automated testing workflows. The mini-project will involve deploying a small application to a cloud provider, and for the group project, students will containerize their full-stack application and establish a CI/CD pipeline. At this point, the group project should be fully deployed, functional and need only minor tweaks.

3.2 Assessments and Grading

Student performance in the course will be assessed through a combination of mini-projects, the semester-long group project, and peer reviews. Each mini-project will account for a portion of the final grade, with their purpose being to make sure students engage with and understand each topic before progressing. The semester-long group project will serve as the main assessment, as it will require students to integrate all course concepts into a fully functional web app with grading based on functionality, design, best coding practices, and collaboration.

Peer reviews will be conducted at each stage of the group project, where each member will evaluate each other's contributions, code quality, and overall effort to help the group. Feedback gathered in these reviews will be factored into each student's individual grade. The students must also maintain documentation of their projects to simulate industry procedures, and this will be graded at the group level. The final deliverable will be group presentations of their finished applications, which will showcase the app's main features and make up the last portion of their grades.

4. ANTICIPATED RESULTS

The proposed course is expected to significantly enhance students' readiness for industry and modern software engineering roles by equipping them with practical hands-

on experience in full-stack development. By working on structured mini-projects and a semester-long group project, students will develop a portfolio of work demonstrating their ability to use tools and frameworks that are standard in industry. This hands-on experience will give them a competitive edge in technical interviews and job applications, as they will have projects to showcase and experiences to talk about. In all, this will help to bridge the gap between coursework and industry. Additionally, the emphasis on collaboration, peer code reviews, and technical documentation will prepare students for professional team-based software engineering environments. Through continuous integration, deployment, and monitoring practices, students will gain exposure to workflows commonly in industry, making it easier for them to contribute effectively once in a professional setting.

5. CONCLUSION

The introduction of a full-stack web development course at UVA has the potential to bridge the gap between theoretical computer science education and the practical skills required in modern software engineering roles. By providing hands-on experience with industry-standard tools such as React, Node.js, and Docker, this course will give students the technical skills and collaborative experience necessary to succeed in today's job market. In addition to these technical skills, the structured, project-based approach will foster problem solving, teamwork and adaptability, ensuring that students graduate with a finished full-stack application and an understanding of its development cycle. Implementing this course will enhance student readiness for industry roles while also helping to align UVA's curriculum with the evolving demands of modern software development. This will be a crucial step toward making UVA's computer science program more relevant, competitive, and valuable to students seeking careers in software engineering.

6. FUTURE WORK

While this proposal lays the groundwork for a full-stack web development course, various steps remain. The immediate next phase involves developing detailed course materials, including structured mini-projects, assignment writeups, grading rubrics, lecture content, and supplemental resources. In addition, it will be necessary to obtain approval from the relevant academic committees and faculty members in order for the course to be integrated into the curriculum. Once approved, a pilot offering of the course would occur. This would provide insights into student engagement, workload, and any areas that may need to be refined. Gathering feedback from students, industry professionals, and instructors will also help refine the course content and structure. Lastly, to ensure the long-term success and relevance of the course, continuous updates will be required to keep up with industry trends. Web development is always evolving, and integrating new frameworks and tools will help the course remain valuable over time.

REFERENCES

Allison, J. (2023, August 7). Choosing computing curricula: Industry needs vs education constraints. The Academic. https://theacademic.com/industry-needs-vs-education-constraints/

Saurabh, S. (2024). How do you teach web development to others? LinkedIn. https://www.linkedin.com/advice/0/how-do-you-teach-web-development-others-skills-web-development

Stack Overflow Developer Survey (2021). Stack overflow. https://survey.stackoverflow.co/2021?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2021