

Education in Modern Web Development
for the University Community

A Capstone Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Jason Giroux

December 7, 2022

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Jason Giroux

Technical advisor: Briana Morrison, Department of Computer Science

Technical Writing Instructor: Rosanne Vrugtman, Department of Computer Science

Education in Modern Web Development for the University Community

CS4991 Capstone Report, 2022
Jason Giroux
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jag8thv@virginia.edu

ABSTRACT

Modern and relevant education in web development is required for many high-paying jobs, yet it is not easily available at UVA. To address this need, I volunteered as an assistant instructor for a 10-week web development course in fall 2021. Source, a Skills Course at the UVA CIO Forge, aims to develop industry skills using hands-on, project-based learning. The course focuses on the React.js user interface library but also includes an exploration of fundamental web technologies; Firebase, a Backend-as-a-Service (BaaS); and Git/GitHub version control management systems. At the end of the semester, students built and presented their own portfolio-ready web application projects. To better serve students, this program should delve into further depth about fundamental web technologies and emphasize the architecture of software.

1. INTRODUCTION

Software is everywhere, and only grows more prevalent in our lives as time goes on. A current high-end production automobile uses “100+ million lines of code” for sensors, firmware and drivers, operating systems, and algorithms for motion control [10]. The Android operating system code base runs on “12-15 million lines” of code [5]. Such complex software in our pockets is used more widely than ever, with 97% of adult Americans owning a phone and 85% owning a smartphone [18]. It is also used widely for entertainment. For instance, in 2021, 65% of adults in the U.S. had played video games at least once [4]. In 2022, software is used broadly for productivity, entertainment, communication, and business.

Web programs have facilitated much of the widespread involvement of software in our lives. We now communicate online through email, online chat, and video calls. We consume music, movies, shows, and books from online sources. We write, create art, and edit images and videos online. We shop and bank with web browsers. Because of the capabilities of web technology, 93% of U.S. adults used the internet in 2021; only 52% used it in 2000 [17].

Thus, there is massive and growing demand for web developers and software engineers. Web developer and digital designer occupations are projected to grow in employment by 30% between 2021 and 2031 [24]. Software developer, quality assurance analyst, and tester professions are projected to grow in employment by 25% over this same period [25]. Keeping up with the rise in demand and pay for such professions, computer-related majors are currently among the top 10 most popular at the University of Virginia (UVA) [26].

UVA therefore has a responsibility to provide students with the resources to learn to create this software. Students interested in web development careers must be capable of creating reliable, scalable, and secure websites and web applications.

2. RELATED WORKS

Opportunities for UVA students to learn web development skills include courses and online resources. UVA BSCS majors must take CS 3240 Advanced Software Development Techniques [27]. In this course, students develop a full-stack web application that does not use modern technologies and architectures. CS 3240 does not require or expect students to use a frontend framework [13]. UVA also offers the course CS 4640: Programming Languages for Web Applications, in which students develop with more modern full-stack architectures. The course was not offered in the fall 2022 semester [1] and will not be offered during the spring 2023 semester [2]. CS 4640 enrollment was also limited to a single section with 156 spots during the spring 2022 semester [3]. Such courses must be modernized and made more widely available. UVA’s Library offers access to O’Reilly for Higher Education, a database with ebooks and online courses about modern technologies [23]. This resource has no enrollment limits but involves no in-person instruction or support.

Other universities have explored ways to provide web development education for their students. Stanford University offers CS142: Web Applications, which covers material about “markup languages, scripting languages, network protocols, interactive graphics,

event-driven programming, and databases” [12]. The course centers around the modern and popular “MERN” (MongoDB, Express.js, React.js, and Node.js) stack, and focuses on class projects worth 55% of class grades [11]. However, the course’s spring 2022 slides use React.js class components [20] rather than the React.js team’s recommended way of creating components: functionally with hooks [7]. Additionally, certain course code snippets use outdated JavaScript language features such as the “var” keyword for variable declaration. The more modern “let” and “const” keywords were introduced to the language in 2015 [19].

3. COURSE DESIGN

To account for UVA’s shortcomings, I volunteered as an assistant instructor for a 10-week, in-person web development course in fall 2021. Source, a Skills Course offered by UVA CIO Forge, was designed to prepare UVA students for careers in software engineering, especially in web development. This was accomplished primarily through a focus on fundamental frontend web technologies, the React.js frontend library, source control using Git, accessing external data using APIs, and an introduction to backend technologies using Backend-as-a-Service Firebase. Besides an overview of these technologies, the course focused on teaching through project-based learning. The main project students worked on throughout the course was a Twitter-like application. Later, students were free to choose a topic for a final project.

The course was structured into two meetings per week. One meeting was three hours long and involved the entire class, while the other meeting was a half-hour lab section.

3.1 Course Structure and Philosophy

Key philosophies that shaped this course included those adopted organization-wide by Forge. All skills courses at Forge run for ten weeks and avoid the turbulent beginnings and ends of UVA semesters. Courses also usually involve a 3 hour-long primary meeting and an additional lab section per week. A time commitment of 3-4 hours per week is intended to fit into students’ schedules without overwhelming or additionally stressing them. Thus, the fall 2021 iteration of the Source skills course involved a primary 3-hour meeting on Sunday nights; students could choose a weekly lab section to attend for solidifying and covering additional content. Such time constraints limited the scope of information that could be taught but served as encouragement for busy students to make the effort to show up and remain engaged in meetings.

Source, like other Skills Courses, focused on project-based learning [9]. It is the belief of Forge that “fun,

hands-on projects” are a fun and effective way to learn new technologies [9]. Technical projects also enhance resumes, serving as concrete proof of a person’s understanding of technical concepts for prospective employers.

The course staff’s own experiences and philosophies also shaped the structure of the course. Weekly meetings usually involved roughly an hour of lecturing and two hours of project work. Lab meetings included the addition of new topics, the practicing of concepts from class, and project work. The lack of outside homework or studying requirements further limited the scope of information that could be covered in such a short-term class. However, the staff believed that limiting outside assignments and instead providing time for project work in class increased the likelihood that students would spend time practicing and applying class concepts. This also gave students time to meet others, ask questions, and learn in a “community-driven environment” [9]. Course staff believed that this course structure would increase student retention and keep students engaged with course material.

3.2 Fundamental Web Technologies

For the first two weeks of Source, students were taught the basics of front-end web development. This began with an overview of the HyperText Markup Language (HTML), which defines the content of a web page. The first course meeting also introduced Cascading Style Sheets (CSS), which define the styling and presentation of this content. During the second week, the staff gave a crash course on JavaScript (JS), a programming language that adds interactivity to web pages. The course staff recognized that the majority of the lessons on React.js would continue to reinforce JS and HTML concepts, so a rudimentary two-week understanding was ruled sufficient for the purposes of the course. Students were also encouraged to continue to learn CSS to style their course projects, though there was not much more class time devoted to teaching this technology.

Though HTML, CSS, and vanilla JS were not vital for an understanding of many of the course’s later topics, they provided background knowledge for the technologies which all other topics built upon. They are the basic building blocks for many of the innovative technologies that have made front-end web development simpler and more powerful in the past decade. The instructional team believed that frameworks and libraries will continue to change, but it is always helpful for students to have some fundamental knowledge of what such software abstracts away.

3.3 The Course Project

After our coverage of fundamental web concepts, the semester-long project was introduced. To reinforce information throughout the semester and let students create portfolio-ready web applications, students built upon their own Twitter-like applications. Twitter is highly interactive on the client-side, perfect for practicing front-end web concepts with React.js. Twitter's functionality was also familiar to most students, so students had intuition about the functionality of the final product they would be creating throughout the semester.

3.4 React.js

The bulk of the course covered React.js, "a JavaScript library for building user interfaces" [6]. This library helps organize frontend code using components. Each React.js component can encapsulate application functionality, HTML content and structure, and CSS styling. React.js was chosen as the frontend focus of this course because of its conceptual similarity to other frontend frameworks and libraries, its popularity, and instructor familiarity.

Most modern frontend applications are built using a library of some kind. Other popular frameworks include Angular, Vue, Svelte, and SolidJS. However, many of the concepts learned by studying one of these frameworks generalize to all others. A person with sufficient knowledge of one-way data binding or organizing applications into components in React.js, for example, would be able to apply this knowledge to most other frameworks and libraries.

React.js is also one of the most popular and in-demand frontend technologies. It is the sixth most-loved web framework or technology in Stack Overflow's Developer Survey, but the second-most loved front-end web framework [21]. The Survey also ranked React.js as the most wanted framework or technology for the fifth year in a row. Due to its popularity, there are many outside software libraries and packages created by the React.js community to solve common problems and add to React.js' default functionality. Its popularity also indicated that students were likely to find answers to their questions online, since their errors and issues were often encountered previously by other developers. Students with React.js knowledge were also able to put an in-demand skill on their resume due to its desirability among prospective employers.

Of all frontend web frameworks and libraries, the three course instructors were all most familiar with React.js. The instructors had learned React.js through Forge's courses, outside documentation, and online materials. Thus, the course was likely to have the most complete

and accurate information for students by choosing this library.

3.5 React.js Fundamentals

After covering front-end web basics, the course pivoted toward the primary course topic, React.js. Seven of the course's meetings were dedicated to teaching this technology. The course first introduced the concept of the component, a way to encapsulate and reuse content and functionality. Later, instructors discussed state, arguably the most important topic for frontend web development using React.js. State is the current condition of an application; it is the data that changes during an app's lifecycle. The current text in an input field, whether an application is in dark versus light mode, and the user's login condition are all examples of state. With this knowledge, students were able to create local "tweets" and display them in their Twitter projects.

3.6 Side Effects and External APIs

After students gained some comfort with state, the course pivoted toward the use of data fetched through external Application Programming Interfaces (APIs). Students added small widgets to their applications for displaying weather information, stock information, or other data to practice accessing and using APIs.

3.7 Backends with Firebase

After adding data from external services, students learned to use their own data sources for their applications. Firebase, a free service offered by Google for hosting NoSQL data in the Cloud, was used to achieve this goal. Students learned to create, read, update, and delete this data from their applications. Students were therefore able to persist their Twitter-clone tweets beyond single sessions in their applications. The course staff were not able to introduce more complicated backend technologies, since this topic would have required significantly more time than the course permitted.

3.8 Routing, Deployment, and Source Control with Git

Throughout the course, there were other miscellaneous topics students learned. After the course's coverage of React.js state, students learned about adding multiple pages to their single page applications (SPAs) using React Router. This library allowed the Twitter applications to have different pages for profiles, individual tweets, and global tweet lists. Students then learned about saving their code's versions with Git and hosting their code on the cloud with GitHub. These technologies were chosen because of their widespread use: Stack Overflow's Developer Survey listed Git as the most popular version control system [21]. GitHub is the most used version control host, with users including

“Google, Facebook, Twitter, and... Microsoft” [14]. By uploading their code to the cloud, students were also able to deploy their applications for free using hosting services Netlify and GitHub pages.

3.9 Final Project

In the final weeks of the course, students were given the freedom to create projects of their own choosing. They were encouraged to think creatively and apply all the concepts they had learned in the course.

4. RESULTS

At the end of the course, students had created two portfolio-ready projects. They had also developed the skills required to create complex web applications for future projects and employment.

Students who came to class meetings on a weekly basis created twitter-like applications and a variety of final projects. The social media applications gave users the ability to view, create, update, and delete “tweets” that were persisted to a cloud database. Students also deployed their web applications, allowing them to share their creations with friends, family, and potential employers. Final projects often reflected the interests of their creators. One student created and styled a landing page for a fictional company. Another project displayed the quotes of a popular music artist, fetched from an external API. One student presented a widget for searching for and displaying global air-pollution data.

Through the creation of tangible projects, students also developed many of the traits required for the successful creation of software. Upwards of two hours were dedicated to project work during most class meetings. Though students were permitted to request help from instructors and peers when they encountered blockers, they were asked to spend some time tinkering and searching for answers on their own. This forced students to debug and find answers online. Students often also had to learn information on their own. Instructors could not cover every detail required for creating complex applications. This forced students to learn to read documentation, watch and read tutorials, and find the learning strategies that worked best for each individual. Students were also encouraged to find new libraries to add functionality to their projects, weighing the tradeoffs between technology choices.

5. CONCLUSION

By volunteering as an assistant instructor during the fall 2021 iteration of the source web development course, I empowered students with the skills required for high-paying jobs in web and software development. Students learned the basics of frontend web development and the fundamentals of the most popular frontend UI library,

React.js. They created multiple portfolio- and resume-ready projects, giving them an edge for employment opportunities over students that lack personal projects using modern technologies. Students improved many of the intangible skills required for success in web and software development, including the resourcefulness to fix bugs and find existing solutions alone.

6. FUTURE WORK

If this course were run with the same time constraints, I would recommend a few changes to its pacing and curriculum. First, I would assign a single reading before the first course meeting introducing HTML and CSS. This would give students a head start on course content, allowing HTML, CSS, and vanilla JS to be introduced in the first week. The time saved would allow for a still-abbreviated but more in-depth introduction to backend web development later. The time dedicated to teaching state in React.js would also be modified. Instructors would still dedicate three weeks of instruction to state, but more of this time would be spent discussing some of the theoretical concepts related to the effective management of state. Students would learn more about one-way data flow, lifting shared state to parent components, and designing user interfaces by “thinking in React” [8].

Ideally, however, the course’s content would be spread into one or two computer science courses at UVA. This course would be easily available to university students, with high enrollment capacities and course sections available every semester.

The new course would cover Source’s existing topics and the discussed enhancements. However, the course would begin by delving deeper into HTML, CSS, and vanilla JS. A more in-depth background in these technologies would provide more context as to why there is a need for frontend frameworks. After discussing JS, the course would also introduce TypeScript, a strongly typed superset of JS that helps prevent many common errors in JS development. TypeScript is a popular tool for many kinds of JS projects, sitting at 30 million weekly downloads on JS’s package manager, NPM [22].

The course would then shift its focus to backend and full-stack web development. Students could learn the basics of Node.js, a “cross-platform JavaScript runtime environment” [15]. With Node.js and a backend framework such as Express, students would be able to create fully featured backend applications [16]. To persist data, students would still learn the basics of NoSQL or SQL databases. Finally, students would learn about creating secure web applications, including the

concepts and implementations for different authentication and authorization strategies.

Though this new course would still take a semester or two to complete, the incentives of a new course would likely cause high retention, student involvement, and motivation. Source requires students to take time out of their schedules to learn material that is not required for the completion of their degrees. If the content were taught in a university course, many of the features that keep students motivated in other courses would exist. Students would know that the work done during the course would contribute credits and count toward their grade point averages. Computing majors and minors would also fulfill requirements for their programs and degrees. The course would fit naturally into regular student life, rather than distracting from it. Instructors would be able to assign readings, lecture for longer each week, and ask that students complete homework assignments.

REFERENCES

- [1] Bloomfield, L. (2022). UVa Class Schedule - Complete Schedule for the Computer Science Program - Fall 2022. Retrieved December 6, 2022, from <https://louslist.org/page.php?Semester=1228&Type=Group&Group=CompSci>
- [2] Bloomfield, L. (2022). UVa Class Schedule - Complete Schedule for the Computer Science Program - Spring 2023. Retrieved December 6, 2022, from <https://louslist.org/page.php?Semester=1232&Type=Group&Group=CompSci>
- [3] Bloomfield, L. (2021.). UVa Class Schedule - Complete Schedule for the Computer Science Program - Spring 2022. Retrieved December 6, 2022, from <https://louslist.org/page.php?Semester=1232&Type=Group&Group=CompSci>
- [4] Clement, J. (2022, May 13). U.S. video game player share 2021. Statista. Retrieved November 15, 2022, from <https://www.statista.com/statistics/499703/share-consumers-ever-play-video-games-by-age-usa/>
- [5] Desjardins, J. (2017, February 9). How Many Lines of Code It Takes to Run Different Software. Business Insider. Retrieved November 15, 2022, from <https://www.businessinsider.com/how-many-lines-of-code-it-takes-to-run-different-software-2017-2>
- [6] Facebook Open Source. (n.d.). React – A JavaScript library for building user interfaces. Retrieved November 15, 2022, from <https://reactjs.org/>
- [7] Facebook Open Source. (n.d.). Hooks FAQ – React. React. Retrieved November 15, 2022, from <https://reactjs.org/docs/hooks-faq.html#should-i-use-hooks-classes-or-a-mix-of-both>
- [8] Facebook Open Source. (n.d.). Thinking in React. React Docs Beta. Retrieved December 6, 2022, from <https://beta.reactjs.org/learn/thinking-in-react>
- [9] Forge Academy. (n.d.). Fall 2022 Skills Courses. Forge. Retrieved November 16, 2022, from <https://joinforge.co/courses>
- [10] Kochanthara, S., Dasjuren, Y., Cleophas, L., & van den Brand, M. (2022, October 17). Painting the Landscape of Automotive Software in GitHub. MSR '22: Proceedings of the 19th International Conference on Mining Software Repositories, 215-226. Web of Science. <https://doi.org/10.1145/3524842.3528460>
- [11] Luong, R. (n.d.). CS142 Course Information. Stanford University. Retrieved November 15, 2022, from <https://web.stanford.edu/class/cs142/info.html>
- [12] Luong, R. (n.d.). CS142: Web Applications (Spring 2022). Stanford University. Retrieved November 15, 2022, from <https://web.stanford.edu/class/cs142/index.html>
- [13] McBurney, W., & Sherriff, M. (n.d.). Project Information. CS 3240. Retrieved December 6, 2022, from <https://f22.cs3240.org/project.html>
- [14] Metz, C. (2015, March 12). How GitHub Conquered Google, Microsoft, and Everyone Else. WIRED. Retrieved November 16, 2022, from <https://www.wired.com/2015/03/github-conquered-google-microsoft-everyone-else/>
- [15] OpenJS Foundation. (n.d.). Node.js. Retrieved December 6, 2022, from <https://nodejs.org/en/>
- [16] OpenJS Foundation. (n.d.). Express - Node.js web application framework. Retrieved December 6, 2022, from <https://expressjs.com/>
- [17] Pew Research Center. (2021, April 7). Demographics of Internet and Home Broadband Usage in the United States. Pew Research Center. Retrieved November 15, 2022, from <https://www.pewresearch.org/internet/fact-sheet/internet-broadband/>
- [18] Pew Research Center. (2021, April 7). Demographics of Mobile Device Ownership and Adoption in the United States. Pew Research Center. Retrieved November 15, 2022, from <https://www.pewresearch.org/internet/fact-sheet/mobile/>
- [19] Rosenblum, M. (n.d.). ExpressJS. ExpressJS. Retrieved November 15, 2022, from <https://web.stanford.edu/class/cs142/lectures/Express.pdf>
- [20] Rosenblum, M. (n.d.). ReactJS Introduction. ReactJS Introduction. Retrieved November 15, 2022, from <https://web.stanford.edu/class/cs142/lectures/ReactJS.pdf>
- [21] StackOverflow. (2022). Stack Overflow Developer Survey 2022. Stack Overflow Annual Developer Survey. Retrieved November 15, 2022, from <https://survey.stackoverflow.co/2022/#web-frameworks-and-technologies>
- [22] TypeScript. (2022, November 15). npm. Retrieved December 6, 2022, from <https://www.npmjs.com/package/typescript>
- [23] University of Virginia Library. (n.d.). A-Z Databases: O'Reilly. LibGuides at UVa Library. Retrieved November 16, 2022, from <https://guides.lib.virginia.edu/az.php?q=oreilly>
- [24] U.S. Bureau of Labor Statistics. (2022, September 8). Fastest Growing Occupations : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics. Retrieved November 15, 2022, from <https://www.bls.gov/oooh/fastest-growing.htm>

- [25] U.S. Bureau of Labor Statistics. (2022, September 9). Software Developers, Quality Assurance Analysts, and Testers : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics. Retrieved November 15, 2022, from <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>
- [26] U.S. News & World Report. (n.d.). University of Virginia Academics & Majors - US News Best Colleges. USNews.com. Retrieved November 15, 2022, from <https://www.usnews.com/best-colleges/uva-6968/academics>
- [27] UVA Computer Science Department. (n.d.). Computer Science Undergraduate Advising. University of Virginia School of Engineering and Applied Science. Retrieved November 15, 2022, from <https://engineering.virginia.edu/departments/computer-science/computer-science-undergraduate-advising>