



**Minutes of the Web Development Program
Advisory Committee Meeting: Fall 2023**

Ethan Wilde, Web Program Coordinator
ewilde@santarosa.edu

Date: December 15, 2023
Time: 5:00pm (Pacific time)
Place: Online conference via Zoom; Zoom Meeting ID is: 698 266 936;
<https://santarosa-edu.zoom.us/j/698266936>; telephone dial-in: +1 669 900 6833.

Members in Attendance:

- Brian Kreck
- Terri Gutierrez
- Cole Lewis
- Kiko Taganashi

Members Absent:

- None

Faculty, Staff, and Administrators in Attendance:

- None

Was there a quorum?

- Yes

Transcript:

Ethan Wilde:

Welcome, everybody! It is that time of year again. We're back for what our annual cycle of meetings. I appreciate everybody who was able to jump in for our spring semester meeting to make quorum. I'm really grateful to all of you for showing up today. Thanks for being here, everybody.

I'm going to go ahead and call the meeting to order. Here we go. We have today with us Terri Gutierrez, Brian Kreck, Cole Lewis, and Kiko Taganashi. We don't need a proxy today, because we're all here. Bravo! That's right on. We already did brief introductions, so I'm going to cruise past this slide. It doesn't look like we have any guests from Career Ed joining us today. I'm sure they're all getting ready for the winter holidays. I've got two slides sitting here for announcements and public comments. I think I'm going to jump straight to public comments. If you all don't mind, I would love to pick your brains about what's going on in your professional world. Maybe we could take turns going around the virtual table here. Because these meetings are not supposed to be so much me talking all the time, although I do

have an important topic to share later on in the agenda about a possible new certificate and degree that we might be able to offer with your support. But I'm curious what's going on in the real world outside of the walls of the junior college and maybe I can just use the participant panel here. Cole, would you mind starting us off and telling us what's going on in your professional world? What's new? What is standing out to you in in the field, and how are you dealing with it?

Cole Lewis:

Well, of course, AI tools. We've been utilizing AI as much as possible to make our jobs more efficient. One of them is Codium AI, that's been really nice. It's an automatic unit test generator. You just install it into VS Code, or whichever IDE you're using. It goes function by function. You can click to test the function. And then what it will do is analyze it, provide unit tests, edge cases, and then it will also provide a human readable version of the function. And it gives you code suggestions on how to improve it. So that's been nice and has really sped up the unit testing.

Ethan Wilde:

Is it a subscription platform that you guys are using?

Cole Lewis:

No, it's free for now. We have also been using Sonar Cloud. That's been around for a little while. It's an automated code reviewer. That's checking for you know, complexity of code, or what whichever configuration that you'd like to set up for it. Just to make the code cleaner. We have just been exploring these tools and having the teamwork through them and see which ones will work for us. Also, we have mainly been seeing a lot of layoffs. My team has been inheriting projects that are new to me. So, I have been working in in different areas and trying to figure things out. What the previous people had worked on. Seeing if there's any documentation left that we could learn from and trying to mitigate production issues.

Ethan Wilde:

Wow, that's quite a bit. So, you're taking on projects that came from other teams that have left.

Kiko Taganashi:

Are they all similar types of projects?

Cole Lewis:

No, just totally varied. There's a Salesforce project that is new for us. You know, we're JavaScript developers and we're usually working in React and Node.js in AWS. So, we don't have any experience with Salesforce. There's Oracle and SnapLogic pipelines. Entire teams were wiped out. It's pretty bad.

Ethan Wilde:

Wow! That's interesting and troubling news to hear. Kiko, what's new in your world.

Kiko Taganashi:

Well, my company's changed its name and we're now mostly in Chile. I'm on my tenth year there, doing JavaScript coding for electronic medical records. It's not really EMR, but I'm working on a templating engine that we're continuing to use for more than five years now. We've got a prototype like I said before. It's a prototype engine that builds production code and makes it easy to change. It's all cloud based. We've actually just started one of the newer things we've done this year. We're starting to move all of our stuff to Azure static web apps using GitHub, which is really nice. It's much cheaper, which is the reason we went that way. But the nice thing about this is that still are using SQL Server for our stuff. But we built a separate API that we can use. We're really not using a lot of the logic parts of as your static web apps, which is file-based, or pseudo-file based. It is all really nice. You just post your stuff, and it updates your production websites. You can have multiple production websites change the code, move stuff back and forth with GitHub to different parts. Then you're publishing. So, you can set up the same

thing where you've got an unstable, a staging area, and then you've got your production systems. You can move your stuff through there and then make it public.

Ethan Wilde:

When you guys picked Azure, did you look at other cloud providers?

Kiko Taganashi:

Well, I've always been using GitHub. I pushed our company to go onto GitHub, because it's just easy to track with multiple people. I'm basically working at a company that's remote. Everybody's remote. We have no office anymore since COVID. The only offices that have people are in Mexico City and Santiago, Chile. Those people are the boots on the ground. We're just a headless horseman, you know. I haven't been in office since 2020 for the good or the bad of it. Collaboration in person is better than being on Skype or Zoom. But I've been doing it since Skype version one. So now it's second nature. It's just not the best. When you have to get things done, it's much nicer to brainstorm in person, right in an office. The other half of it is that you get to go off on your own. What I like to do is come up with an idea and then go, "Here are the methods that I could do this with." I propose it to people, and they if they're on board, then it's my challenge to actually make what I think is going to work. It's a good pathway. That's where I'm at. I've been doing the same thing. It's just at the next level. And actually, unit testing is one of those things that we don't do that we should be doing. Because we're running on a solid platform that is running itself, when you get an error, you see it right away. Fix the error. It fixes it everywhere. So that's been the great thing about it. We're down to almost no errors. I've got one ticket that has been outstanding, and it's just a future item. It's been a long one, though, building an offline website online. So that's very complicated. I'm caching using the Google version of an offline website or a mobile app. But it's all Web-based. It's updatable. It's good. That's where I'm at.

Ethan Wilde:

Let's hear from Brian. Brian, what's new in your world?

Brian Kreck:

So not much different. We're not planning to return to the office anytime. At this point I think we're probably going to stay with everyone at home. I hate it. But the costs are just so attractive at the moment. I have a co-location facility. I have certain people that I know are vulnerable that are really just not happy, so I go and meet with them, and we have beers. I think I'm starting to navigate it. But honestly, management for me of remote teams is what it is. I've never trained to be a manager to begin with and didn't really want to be a manager. I felt insecure. But I think we're heading there, and I feel like I'm starting to get a little bit more comfortable with it. There's AI tools right now. I'm just absorbing everything. We use it for certain things. We've used a couple platforms that you guys have mentioned. But I'm still struggling to figure out how to make it work. I'm looking for some more. I play with Selenium and some visual unit testing stuff. I don't know if you guys have ever played with that. We have roughly about 300 websites. We're a small boutique company with about 300 websites. What randomly happens is that someday some plugin or something doesn't work just right. Or a thing that visually showed up doesn't show up anymore. Or whichever client changes the thing we didn't anticipate affecting anything. Then a little block on some page somewhere changes. What I've been really searching for is something that finds this and alerts us. I don't know if you guys use a service that pings websites. We do. Every x number of minutes. It downloads essentially the contents of the page and might check the page for certain phrase, some markup or some blurb. With this tool, we essentially write a special hidden page. The payload is tiny. That one page runs a bunch of checks and then returns a magic string like and that's the one you're monitoring. All of our websites check a couple of other things. You'll find five lines of code saying, "Hey, I check the database. I check this. I check that everything's running great. All tests pass." It's been a good thing for us. We've been in this for 15 years. We'll use a monitoring service, we'll hit it. But what I really been looking for is some way to automate a visual check and get back a message that says something like, "10% of this page has changed," where something is visually comparing a

before-and-after scenario. We did have a tool that would do this. Super helpful. You guys may use Gulp or Grunt, or one of those task runners of some sort. Imagine, suddenly, some big Node modules have been updated. And you say, "all right, I'm going to re-review this. I'm going to push it all up. It's probably fine." Suddenly, on page 20 of something, suddenly something changes right? We used to have a service, that has been discontinued, that would do a pre-render of the page, and then we push our changes, and then we just hit a button and bang, this thing would pop up and say, "Hey! On this page this tiny little thing changed." It was like a visual diff. I just want to create a super high-quality product. I just constantly want to be able to deliver top-notch service to all my customers. It's really embarrassing when you push changes and then suddenly, they go, "Hey, on this page this thing's not showing up anymore." So, I'm just constantly looking for ways to automate knowing that something has happened, and it doesn't have to be anything other than a flag. Something big has changed. So little things like that. I'm hoping this is exactly what AI is for. Right? Just download all the pages, take a look and then tell me what's changed. Just call out things that have changed in a major or minor way. We're trying to improve stuff a little bit. I've complained about hiring people forever with you guys. Probably the last 10 years. It is much better right now. I think while the layoffs have happened, a lot of people getting laid off are getting snatched up quickly. I do feel like that the junior people are really struggling, those in their twenties. They're really struggling. But some other people I've talked to most of them were not unemployed for that long, those that have that have recently switched jobs. I've had more opportunities for getting people of all different types. Even on the winery side. So, as many of you guys know, I have a winery as well. And I've just landed one of my favorite people I've ever had the opportunity to hire, and I think he's amazing, with 15 years of experience. I've struggled to get people like him in the past. That is somewhat exciting for me. I'm still getting tons of opportunities. Tons of business. That sort of thing. All of that seems really good. Like I said, the junior people are struggling. I do feel like a lot of senior people are able to make use of tools that previously a junior person just had to do 4 hours of grunt work to address. Now there's other options. They're not hiring as many of those junior people.

Ethan Wilde:

That's really interesting. That makes sense.

Brian Kreck:

That's the quick summary. Typically, we'd be speculating about what's on the horizon. Right now, I honestly feel this whole industry is shifting. Right? It's been a rocket since I first got into this in 1997 with my first website. 1996 was my first website. I came out of UC Davis. We were learning what Netscape was. Then I came back, and I joined the family winery. Everyone was saying, "There's this website thing. Can you do this?" I said, "Oh, yeah, sure." Ever since then, obviously, things have gone quickly. But right now, polishing of my crystal ball, it is getting really rough. I do not know what this industry is going to look like in five years or 10 years. I never felt like I really could predict anything. But it's really rough right now. I do not know if I will have a job in five years. What I do I think will be easily replaceable. I do not believe this is going to last. Previously, I was always able to pivot as those things came up. Like Wix and Squarespace come up and it's no big deal. Who cares? Clients aren't able to use those tools most of the time. I am very curious about where things are going. I'm an old guy now, and so essentially, it's less scary for me now, than if I was 28 right now, or something like that. I would feel differently.

Ethan Wilde:

Great observations.

Cole Lewis:

I think it's funny. I've thought about this. Obviously, everyone has recently. In a way it feels like we're doing math without a calculator. And pretty soon, people are going to laugh and think, "You wrote this code manually?" It's starting to feel a little bit primitive.

Brian Kreck:

Yes, that's interesting. There's a developer that I am friends with. He first started working with me, and we would get together and talk. We'll look back and say, "We did this. We did that. Where is it now?" It is weird. You write a book, and 50 years later, your book is still readable. While so much of what we do in web development is ephemeral.

Kiko Taganashi:

It's going to go away, like a temporal painting.

Brian Kreck:

I envision it that way. I think that's a helpful way to look at it. You're building something beautiful that's not supposed to last. It's here just for now.

Kiko Taganashi:

One of my coworkers has a thought about that. They speculate that technology like this was here before on our same planet, but we don't know about it because it was all electronic. There's no records, right? The only things we see remain are the physical.

Ethan Wilde:

I think about that too.

Brian Kreck:

Honestly. 50,000 years ago, if there was something like that, I don't think we would know about it. It's possible.

Kiko Taganashi:

They keep finding older and older pieces of history, and say, "Oh, we didn't realize that 100,000 years ago this existed, before we thought people were here."

Ethan Wilde:

I feel like all of us have been on a similar trajectory. I completely relate to it. Everything that everybody's saying about the uncertainty of the next five years in this field resonates with me.

Kiko Taganashi:

As long as we don't completely outsource our imagination, I think we're useful for something. I like that perspective. That's a really good way to think about things, because that's what it comes down to. Even the work that I'm doing, it's not really, at a certain point, the programming that's being done. It is not really even programming. It's more like workflow management. Already, everybody can get online tools that somebody's written before to do some of the things. Just to generate it. It's more about asking, "how can I creatively use what's already here? And how do we make like the design simple?" Because we all know that you can make anything. Just look at Node modules. They're massive. How many hundreds of thousands of kilobytes of code do you need to run a simple page? Right? In the future, things should actually be very smart. Then there's very little code. If you think about things, then maybe you don't have to do that at all. I think that novel approaches will still continue. For how long I don't know, but as long as we don't outsource our imagination and our creativity, it's still there. You can tell a computer to draw an amazing picture. And I started originally as a graphic artist and got into programming, because it was a way to make some money. But you know, you can tell a computer to make something more amazing than what you could paint yourself. But that doesn't make it unique. You can still paint something unique. In the creative world of computers, we still have something to offer. People coming up, new people, still have something to offer. It's just important to pivot in the way you would think about things, the way you would do something. I think that's the next generation of things. How do we best use what we have? Because maybe those old jobs will disappear, the ones that we

thought were so amazing and creative. But the next step is, “well, now, how do we use this? And how is it useful?” Because we can make a lot of things, but what is actually going to work? You could make a bicycle with triangular wheels, but it's not going to make you go down the road, or does it go better than one with round wheels?

Ethan Wilde:

Great perspective. Could you say more about your project that you were talking about, using GitHub with a static site generator?

Kiko Taganashi:

Yes. They are static web apps. We have 3 facilities running on it. We're starting to run them across Chile. We have a master schema on our SQL backend. As we start to update everything on our development platform, we push them out and they will replicate. It's basically like we're replicating the same thing. In the old world of hospital technologies, everybody wants everything in house. They want to own everything. The government wants to make sure that the software is generated inside the country. What we're doing is we're moving. Microsoft is building a data center in Sao Paulo. It's not done yet. We're still running everything from up here until that gets built. They're becoming more and more of a partner with us. I'm not sure how that is, but they're excited that we have business for them. Once that data center is in there, we are migrating to it. Currently, the way we're working is we have partners. We have sales partners and implementation partners, because the company needs to be in Chile, Argentina. In other countries, you need to have local partners, or you don't get the business. Right now, we've got some good partners, as far as implementing and selling. It is like 500 times bigger than our five people in Northern California building this stuff. But it's a good place to be, because we've got something that's working. We can continually work on revisions, making it better. It's really about how to get it into places. It would be nice to have more people here. We used to have 20 people here in the Santa Rosa area. Now we are down to five. But we've got 20 something people in in Mexico as a team. And along with our partners down in Chile, there's probably a good 40 or 50 people. We just partnered with a bigger company, and they've already got the business. They sell stuff everywhere, but they don't have the business that we have. They're interested in moving into the healthcare world. Even with all of the stuff that's going on with groups like Amazon that has their AI health platform. Everybody's coming up with these tools. But does it work for you? Is it built for you, right? That's the place we're trying to be at. Do the client-centric thing. We've built the box. We can configure the box and the box can look the way you want it to. What do you use it for? How does it work? Who uses it? What teams use it? Right? This is how we can make this. In the end, it's all really just forms that store information and display graphs. Nothing too exciting. But it's the user experience, versus the interface. The interface works pretty well. The experience of what the user gets is most important. I'm a nurse. I come on shift. What do I have to do? I'm a doctor. I don't really care about what the nurse does. The nurse does all my paperwork, basically and I do my surgeries. The jobs are different, even though they're using all of the same data. Right? How do you break it up? And that's the creative thing we work on versus being the programmer. We've already done the programming. Now, how do we make the program work in a bigger sense?

Brian Kreck:

Yes.

Kiko Taganashi:

That's where my headspace is. You may be doing other things in your life. How do you make it work for the people that need it to work for them? And does it have to be so complicated? We're trying to make it less complicated. Things like integration with SAP and Salesforce. All those people have got tax systems, and risk systems, and imaging and laboratories. I know there's all these tools already out there to convert formats and protocols. It's the job that you could use AI for. Those are all old school. We're using all this old tech. We're just trying to modernize the old technology that's out there in a way that people can use it. And make it easy. It's like reinvention of paper. How do you make paper better? It's in

a hospital system. Because the only downside of paper is that you can't share it with everybody right now. It's very flexible. Everybody loves paper. When you go to Kaiser, or you go to Sutter, the system goes down all the time. Then they pull out the paper. They've got a paper form, but if you've got an electric version of that paper form that does better than the paper, and it doesn't break down, and what it does we can fix it right away, and it doesn't take booting up the system. That's the bonus of being cloud based. That's the bonus of using the browser as the workhorse instead of the server as the workhorse. It sounds like the medical space is ready for this. At least for us in the programming world, that's a good place to go. There's a lot of places where there's still old systems that could be linked into the present and the future. Getting them to do the jobs that people do without making it complicated. Because that's truly the point of using a computer. The computer is there to be the tool. It's like getting in a car and driving to the store. You don't need to check a bunch of boxes to do that. Although if you have an electric car you may have to. But you know what I mean.

Brian Kreck:

One of these days, I'd love to have a beer and talk about when organizations should explore custom software and when not. Through 20-some years, that's been a very constant topic. I think it is a fun one for all to explore.

Kiko Taganashi:

Yeah, it is interesting. I just think about you, Ethan, when you had your CMS software. Eventually you have off-the-shelf solutions for things that once required custom software solutions. Now, people are using Drupal, or using WordPress. Those are all the same tools. WordPress is pretty is great, but it's also a clunky beast.

Ethan Wilde:

On that note of content management systems and other stuff, let's bring Terri into the conversation. Do you want to share anything about what's new in in the world of managing hundreds of CMS-based websites?

Terri Gutierrez:

I think one of the issues that we've actually had at the JC is that we have a lot of users for the different departments that have their own website. And they manage those sites. They manage the content that they'll build, you know, the pages. So, the issue that we're having is it's like the Wild West out there, where they're building these ugly sites, and they don't include us in the conversation. We're trying to regain some control of some student-facing sites that are really important to maintain a consistent branding, a consistent look and feel. That's what we're trying to do. We're trying to regain control. There are people who don't want to give up control. They have a lot of attachment to their sites, so it's difficult. But Leela and PR are in conversation. They decided that they're going to start with a couple of sites. I don't remember exactly which ones, but we're going to start with a few sites to redesign and rebuild with the new layout and the new design. That's what we're working on. As I mentioned, we migrated to Drupal 10. With Drupal 10, we have to switch from the current rich-text WYSIWYG editor to a new version of the editor. We haven't done that yet. We are just testing it on a few sites. There's not a big difference between CKEditor version 4 and version 5. One of the major differences is there's this HTML code button in addition to the source button. We're trying to decide whether we want to make those two buttons available. How will the administrative users receive that? Would they be confused or not? One of the issues that we discovered is that we have some code for accordions in the "kitchen sink" set of resources. The "kitchen sink" is basically the HTML code that one pastes into source. With those accordions, it doesn't work anymore. The editor adds a bunch of paragraph tags to the markup that basically break the JavaScript for the accordions.

Kiko Taganashi:

That's not good. I had to take that out of one of my systems as well. People were making HTML emails for appointments and stuff like that on our system. We had to remove it.

Terri Gutierrez:

Yes. The way that we were trying to get around that is using the new HTML button. I think it's an HTML embed button. If we paste the code into the HTML embed button, the accordions will work. If we paste the markup straight into the source mode, then it just breaks all functionality of the accordions. We're trying to decide what we're going to do with that. It's such a tiny little change, but it has such a big impact.

Ethan Wilde:

That reminds me of what Brian was saying. So, you implement this new rich text editor and suddenly all the UI elements like the accordions that you have ubiquitously everywhere, are suddenly fragile.

Terri Gutierrez:

We only implemented CKEditor version 5 on a few sites so far, just for testing. For one user, I think Corrine, all of the accordions that she had built were broken. We don't know how it's going to affect all the other sites. Unfortunately, I'm assuming there's going to be issues that we're going to have to deal with afterwards. That's going to be a big thing. Also, now we're back working in Bussman Hall. The remodel was completed several months ago, so we're back on site. I'm working two days a week there and then three days remote from home. It's nice that I don't have to be going to the office five days a week.

We also have another migration. We have been using FormStack for our web forms and we are going to migrate all of those forms onto new servers. We haven't started the conversation with our account manager yet, in order to see how that's going to affect our workload. I believe it can be completed in a bulk operation, so hopefully that will not give us any issues. As you may already know, we have a new coordinator of online accessibility. Dana. who was an STNC, working for PR. She's joined our team, and luckily, we have not had any layoffs. We just added a third person to our team to alleviate some of our workload issues.

Ethan Wilde:

You are in a similar realm to where Brian's at, with hundreds of websites. Thank you, everybody, for sharing a little snapshot of what you're all grappling with, and what you're worried about on the horizon as well. I really appreciate knowing all of that.

I'm going to move us ahead a little bit. I have a little bit of business to share, and a couple requests for voting in our meeting. First, just a quick shout out our student-run web and mobile studio. Our Web Projects class is firing up again in the spring semester in the third week of January. We've got a landing page on the Computer Studies site for the studio. We're currently soliciting folks that need new websites. Our focus is on non-profits and small organizations. They typically can't afford to retain a commercial studio to do the work for them. We have our intake form open. I haven't looked lately, but I think we're stacking up possible projects. We've got three dozen student enrolled already. It's a great class. That class has stayed remote. It went remote during COVID. It used to be a face-to-face class. We have not been able to get students to come back to school for Web Development courses, not that we had many face-to-face sections before COVID. Everything that's done on the computer, people are not wanting to be in the classroom. For example, right now enrollment is open for the spring semester. We have one section of our introductory Web Development 1 class that has a face-to-face weekly meeting. It has three students enrolled in it. We have over sixty students enrolled in online sections of the same class already. So that's the ratio. It's almost 20 to one. I don't have any leverage with my students to force them to come to the classroom. Frankly, the program would fail if we forced everybody into the classroom. The chemistry students all came back because they couldn't do their lab experiments at

home. But for our program, you really only need a Chromebook to get through our program. Everything we do is browser-based. Interestingly enough, because we're fully online, we're getting a noticeable number of high school students enrolling that wouldn't be able to take a face-to-face class. There's a big push for what they call "dual enrollment" so that aligns well. We're also getting students enrolling who are incarcerated. The JC has a strong and growing outreach to the local county jail. So, I now have students, basically young people that are trying to get off of whatever bad road they were on in their lives, that are getting web development degrees.

Kiko Taganashi:

Well, that's great. They can't be on campus anyway. They would not be able to participate if we were in a classroom.

Ethan Wilde:

So that's an update on that front. We do have a new opportunity. In other news, we have a new department chair in our department. Our long-term department chair, Donald Laird, actually just retired. Brian, I know you're familiar with him. He did a great job of keeping our department above water during a lot of really challenging semesters. Our new department chair, Michael McKeever, is our subject matter specialist in network security. He arrived just in time for the school to try to grow enrollments after we were hunkered down, trying to survive the fires and COVID. Now there's some encouragement for us to start new programs. So, we're taking off the back burner a program that I tried to get started with the lead faculty who oversees our transfer-level programming classes. We want to start up a mobile app development certificate and degree that builds on the Web curriculum for hybrid apps, and then adds a couple new classes for strictly native iOS and Android app development. I would love to get everybody's engagement on this topic tonight in the little bit of time we have left. We've got a draft curriculum outline. I just put the slide up for that. It's leveraging the introduction to programming (CS 10A) class which is a four-unit, one-semester class, and much of what we have in web development. This includes our three semesters of basic HTML, CSS and JavaScript that then lead to server-side JavaScript. There we are teaching React and Next.js. We're also using Google's backend-as-a-service Firebase which has got authentication and NoSQL database services. The nice thing about Firebase is that it is all free. We looked at AWS, but AWS would not give us anything guaranteed as free. I have not really investigated Microsoft's offerings, but the Firebase platform is really powerful, well-documented, widely used, and easy to pick up. So that's what we're running within that class. We also have a class on the books that we've never taught that is designed to teach hybrid apps, CS 55.14. Right now, the plan is to use the Ionic framework, which basically lets you wrap up a React project in a native application wrapper, and it exposes all the native iOS and Android APIs to JavaScript. Then we're planning to build two new classes in parallel. One is going to be an iOS-specific class. The other is going to be an Android-specific class. Both will be purely native apps. So those would be the two new course offerings. Out of this curriculum, we end up with a 23-unit pathway which makes us eligible for an advanced certificate or an AS degree. If anybody's got questions about anything I just said, I'm happy to field them. If you've got suggestions that you think we should be considering as we try and put this curriculum together, I'd love to spend some time hearing from you on that. I'm hoping we can vote, with your endorsement, to proceed to submit the certificate and degree in the next semester. It would actually be fall of 2025 before the pathways started. It's almost two years before students would be able to enroll. Any thoughts on this? Do you think this is a good idea for us?

Kiko Taganashi:

I do. I think it's great. This is like I was doing with Cordova apps a while ago. I think with the hybrid apps you could also be doing something with the Electron framework, because that's also desktop based.

Ethan Wilde:

I have been introducing Electron in my game classes. The game engine that we've used most of the time is a JavaScript-based game engine. I show students how to use Electron.js to roll up a desktop Windows

or MacOS app out of their game, which is pretty cool. Visual Studio Code is built with Electron. They all use VS Code. I pull up the slide where I ask everybody, "Do any of you use an app built with Electron?" They're saying, "I don't think so." And then I show them that VS Code is built that way. And they're saying, "What? Wow." Skype is the same thing.

Kiko Taganashi:

Yeah, Skype is. It's amazing when you think about it.

Ethan Wilde:

Totally. I love Electron.

Brian Kreck:

We use Evernote, Asana, and QuickBooks. They're all Electron-based.

Ethan Wilde:

The JavaScript language isn't bounded by the Web browser anymore. Students can publish to a native mobile device, they can publish to a desktop operating system, using the Web technology stack they already know. It blows their minds.

Kiko Taganashi:

Yeah, right? Plus, you're actually able to leverage the system for its native usages. All the native APIs are mapped in there.

Ethan Wilde:

Exactly. It is really powerful.

Kiko Taganashi:

Yeah, I like Ionic. I haven't used it in a while. But there, there's even more to do there than just React. You have got every framework here. They got Angular. They got React and Vue. You could just build something with those. It's pretty neat.

Ethan Wilde:

In the server-side class where we're using React, and Next.js and Firebase, during the last 3 weeks I turn them on to Ionic. They built their first app they compile it for Android or iOS, and they're all blown away,

Kiko Taganashi:

It's pretty fun and very cool.

Ethan Wilde:

So, this is the pathway, the door, the gates have finally opened at the Dean level for us to try and get this through the window of opportunity. The timeline for this is always longer than I want it to be.

Kiko Taganashi:

Are you doing Swift? Are you going to be using Swift on iOS?

Ethan Wilde:

That's the plan. There's a curriculum already built by Apple for a college class teaching Swift. Also, we finally got a new faculty member who is a programming instructor. So, I think we can staff these classes.

Kiko Taganashi:

Wow.

Ethan Wilde:

I appreciate those comments that you shared Kiko and happy to hear from anybody else, if anybody else has thoughts about this.

Cole Lewis:

I was just curious on the reason for having the two separate iOS and Android classes. I suppose that would be Swift or Java, rather than teaching React Native, for instance.

Ethan Wilde:

With the class that's already on the books, CS 55.14, it's written in a way that it's pretty much platform agnostic. Right now, my plan is to use Ionic there. But I think there's an opportunity there to do something like React Native instead, or in addition to Ionic. Maybe split the semester in half and start off with one. Right now, we basically end the server-side, full-stack, class with 3 weeks of Ionic, which is just like an easy place to go after they've been working in React all semester. The mobile app design class, CS 55.14, might have the space to do something like a React Native version of that class.

Kiko Taganashi:

It does make sense to use React in some way, Ionic, or React Native. It really shows that you're using the same tools.

Ethan Wilde:

Totally. I appreciate that feedback. I could definitely make that work.

Brian Kreck:

I think, from the marketing standpoint, back to the industry view of things, we went through a period where all of our clients were looking for a native app. They're like, "I need an app, right?" I got to have an app for whatever, you know, stupid stuff. People were just creating an app that did nothing but using the camera. They're not using GPS or not using like any of the things you might feel like you need to use. I think that users are pushing back, saying, "Look, I don't want to download an app, you're another barrier to entry, right? I don't want to download Acme's app when Acme has a permanent website where I do everything." It's no big deal, because for developers, for your students, there are tons of good app projects out there. But I do think that that the gilded aspect of all apps is fading a little bit.

Ethan Wilde:

That's a great point, Brian. I think that talking about what the best used cases are for native mobile apps is a key part of the curriculum. That is going to be important. Leveraging what's unique, like you said, either camera, GPS, other things that a smartphone lend themselves to that are distinct from just being on any old web browser.

Brian Kreck:

I mean, there's tons that you just can't get access to through the Web.

Ethan Wilde:

Exactly. That's a great point. Thanks for bringing that up.

Kiko Taganashi:

Yeah. Absolutely.

Ethan Wilde:

Shall we move onto action items, and then we can wrap it up and think about holidays, right? I would love to ask for a motion and a second to approve the spring minutes.

Kiko Taganashi:

I move to approve the spring minutes.

Cole Lewis:

I second.

Ethan Wilde:

Thank you, Cole. All in favor?

All:

Aye.

Ethan Wilde:

Okay. Last up is an important milestone for us. A chance to vote to endorse the idea of firing up a mobile app development pathway, both a certificate and degree for the JC students. Can I get a motion for a vote on that?

Kiko Taganashi:

Yeah, I vote yes.

Ethan Wilde:

I got a first from Kiko. Can I get a second from somebody?

Brian Kreck and Cole Lewis:

I'll second.

Ethan Wilde:

Thank you. I think I got a second and a third. Wonderful.

Brian Kreck:

Alright!

Ethan Wilde:

All in favor?

All:

Aye.

Ethan Wilde:

Wonderful! I am extremely grateful to all of you for taking the time. I know we don't see each other very often. Brian, I think your idea of getting a beer sometime with our group would be a great idea. I'll send out an invite for some time between now and late January, when the semester starts up again. It would be a really nice idea to actually be in the same room together. That was the collaboration piece, the real collaboration.

Kiko Taganashi:

My beer might still be good by then. I made some orange honey beer like late summer. If there's still some left.

Ethan Wilde:

Okay, the window is closing for us to meet around Kiko's homebrew. I like that.

Ethan Wilde:

Thank you all. Our students are lucky to have the guidance that you all provide. I think we've got a wonderful distribution of people on this committee that have their hands in all of the important stuff that our students are going to be facing when they go out to the world. So much gratitude to each of you. I hope you all have a great winter holiday season and a wonderful New year. I'll shoot out a note. Maybe some of us can get together sometime early in the New Year and be in person.

Brian Kreck:

Awesome! Do you mind if I drop one little issue, I'd love everyone just to take a peek at in the chat and then talk about it? Essentially, there is a bill in a California State Legislative Committee. There is an amendment to the ADA law surrounding how the ADA applies to websites. I personally find it to be terrifying. The way I'm reading it, the way the analysis is, it would require every almost everyone with a website in California to have a formal ADA audit every year.

Kiko Taganashi:

Oh, my God, it's like a huge new business for consultants. I disagree with that one.

Brian Kreck:

It would be fairly unpleasant, I would say. I have a little tiny summary of some links that I will drop in the chat. That's really important.

Ethan Wilde:

Thanks, Brian, for sharing this information. Have a great evening. Thanks so much for all of your help.

Kiko Taganashi:

Alright! Have a good holiday, guys. Talk to you later.

Cole Lewis:

Good night.