The Curriculum

Web Developer

Basic Digital Literacy

The Basic Digital Literacy module lays the foundation for the participants to be able to use common development tools, identify the technological basics of the internet as well as introduces them to basic workflows developers face in their day to day jobs. By the end of this module a participant should be able to:

- Use the linux bash terminal to navigate, create, view and manipulate files and folders in the operating system.
- Visualize how the internet works and identify various components of a URL
- Use tools such as git and GitHub to implement a versioning history and collaborate on their projects
- Author documents using a lightweight markup language such as markdown

> “
>
> It is easy to shoot your foot off with git, but also easy to revert to a previous foot and merge it with your current leg.
>
> Jack William Bell

UI Basics

The UI basics module provides the participants with a deep understanding of how to create user interfaces for websites and applications as well as employ various development tools to support their efforts. By the end of this module a participant should be able to:

- Plan out and create user interfaces for various screen sizes and use cases using simple wireframes, HTML and CSS.
- Differentiate between various types of content and interactive elements.
- Publish a website and adopt common development workflows using different build tools and frameworks.

> You know what HTML really stands for? How to make love.
>
> Bjarne Sørensen

Programming Basics

The programming basics module is designed to train the participants to approach various logical problems and develop problem solving skills and tools, as well as teach them the basics of programming and debugging from the ground up. By the end of this module a participant should be able to:

- Identify common logical problem patterns and solve them by writing algorithms.
- Create small problem specific programs in JavaScript.
- Understand intermediate programming concepts such as recursive functions, closures, arrays, objects and classes.
- Identify and debug compilation, runtime and logical errors in JavaScript.

> JavaScript is as related to Java as Carnival is to Car.
>
> Kyle Simpson

Browser

The Browser module provides the participants with their first experience in working with browser environments and 3rd party APIs, either locally on the client side or remotely by working with data from a server. By the end of this module a participant should be able to:

- Work with the DOM API to manipulate HTML documents and implement interactivity using JavaScript.
- Split their code into multiple files and use the ES6 module specification to include third party libraries, as well as adapt legacy code to remove libraries.
- Understand the basics of working with asynchronous code.
- Work with external servers and JSON in order to retrieve, send and store data within the browser.

> “
>
> There may be 300,000 apps for the iPhone and iPad, but the only app you really need is the browser.
>
> Jim Balsillie

Node

The Node study module aims to teach the participants about creating programs that work with the operating system, as well as give them a steady ground for creating and publishing their own simple utility programs and packages. By the end of this module a participant should be able to:

- Create JavaScript applications that run using the Command Line Interface.
- Read and write files to the operating system using JavaScript.
- Work with asynchronous code to implement local and remote 3rd party APIs into their programs.

Single Page Application

The Single Page Application module is where participants gain first-hand experience with common architecture patterns facilitated by JavaScript front end frameworks. From starting up their own apps, to integrating a state store in their client side applications, participants gain a perspective on current common implementations in web development. By the end of this module a participant should be able to:

- Create, expand and deploy their own single page applications using a framework, such as React.
- Identify common architecture patterns in the context of client side applications.
- Understand and employ reusable components to create their applications
- Integrate helpful libraries for routing and state management into existing applications or applications created from scratch.

> The Facebook codebase has over 20,000 React components, and that's not even counting the mobile features.
>
> Dan Abramov

Data Server

The Data Server module trains the participants to use common Database technologies and Node frameworks to create a transactional RESTful API service for storing and serving JSON data. By the end of this module a participant should be able to:

- Create an express based application from scratch with 3rd party libraries to handle common use cases such as validation, sanitization and authentication.
- Understand and use basic database concepts to store, manipulate and retrieve data.
- Create and administer MongoDB databases using an ORM such as mongoose
- Understand the role of RESTful API servers in the current web development landscape.
- Deploy server and databases to third party service providers, such as now and mlab.

> A few years ago, it would have been unthinkable to implement server software in JavaScript.
>
> Guillermo Rauch

Full Stack Server

The Full Stack Server module introduces the participants to a common implementation architecture for web applications which relies solely on a remote server for all aspects of the model view controller pattern (MVC). By the end of this module a participant should be able to:

- Understand use cases for cookies and sessions. And implement them in an express application.
- Use templating engines, such as handlebars, to render and serve dynamic HTML views.
- Employ various 3rd party libraries to secure online forms, accept file uploads, implement various authentication strategies and verify user emails.

> “
>
> Complex applications combine different types of problems, so picking the right language for each job may be more productive than trying to fit all aspects into a single language.
>
> Pramod J. Sadalage

Final Application & Project Phase

With the start of this phase, the participants start to apply intensively and work on their final projects at the same time. In order to apply for internships, our participants will attend three workshops: “How to write a good CV”, “How to improve my online presence as an applicant”, “How to succeed in an interview situation”. Our Alumni Management department will support the participants in finding an internship.

After learning the basics of web development, with different technologies and frameworks, the participants will work on a project. They will create an application based on a real-world use case. The participants have to work in small teams and define different roles within the team.

When the final project is finished, the participants prepare a presentation for an event where they showcase their work to the other classes and modern tech companies. The showcase event is a chance for external companies to convince participants to do an internship with them.

> “
>
> Don't watch the clock; do what it does. Keep going.
>
> Sam Levenson

Summary

– – – – – – –

Single	Parallel	(Teaching Model)	
	2	Weeks	Basic Digital Literacy
8	16	Weeks	UI Basics
8	16	Weeks	Programming Basics
3	6	Weeks	Browser
2	4	Weeks	Node
6	12	Weeks	Single Page Application
6	12	Weeks	Data Server
2	4	Weeks	Full Stack Server
	7	Weeks	Final Application & Project Phase

Included during the whole year:

1 Team Building Workshop

3 Application Workshops

Project Management

Hiring Events

Hackathons