	Sub Area Version Control	Topics - branching	Estimated Time	Resources Udacity course for verson control and git: https://www.udacity.com/course/version-control-with-gitud123
Foundation	• croion CUIIIO	- orancning - push / pull - commit creation / templates	0.5	осколу осклосто, чегоот солногаты ун. тирэлимим.инаму.соптооновгченног-сопции-мит-ун-чи 123
		- merge requests		
		- conflicts resolution - rebasing / squashing		
		- reverting a commit - cherry-picking a commit		
		- self code-reviews before submission		
	Programming Language	 - Variables, Data Types, Type Conversion, Loops, Conditionals, Undefined, null, Type Coersion - Functions (expressions, declarations) 	2	- You Don't Know JavaScript by Kyle Simpsons Raafay - Frontend Masters
		- Scopes (global, function, block), - Variable Hoisting		JS Fundamentals (essentials)
		 Compilation vs Interpretation, Debugging Objects (Literals, Constructor Functions, Methods, Nested Objects, dot vs brackets notation for access) 		https://frontendmasters.com/courses/is-fundamentals-functional-v2/
		- Arrays - read docs of common array methods, - Nested Functions, Anonymous Functions, IIFE and when to use it		https://javascript.info/
		- Functions/Code as arguments - Closures		JS in depth: (essentails) https://github.com/getify/You-Dont-Know-JS/blob/1st-ed/README.md
		Object methods and uses of "this" 4 flavours of this, Apply vs Call		In depht JS: (elective)
		- Function "arguments" array, Object.assign - Read about Date, Math, string		https://frontendmasters.com/courses/javascript-hard-parts-v2/
		- Prototypical Inheritance/OOP in JS - Event Loop, Async Flows, setTimeout		Very through JS fundamentals: (elective) https://frontendmasters.com/courses/getting-started-javascript-v2/
		- Async/Await vs. Promises		
	Web Concepts	 Client-servier communication Models (HTTP, Pub-sub and Sockets, Async communication,) Communication of app's different components (such as DB, Server, Frontend, Web) 	1	How HTTPS works: https://howhttps.works/
		Authentication and Session Management Server-side and Client-side Rendering		Session Management: - https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
		- Server-side and Client-side Rendering - Caching (types of caches on BE and FE) (bilal to add) - High-level components of BE application (bilal to add)		How Modern Web Apps Work
		 Browser Understanding (inspection, debugging, profiling etc) (umair to add) Security (HTTPs) (qadeer to add) 		 https://litslink.com/blog/web-application-architecture https://existek.com/blog/web-application-architecture/
		- Misc Topics (User-agent, CORS, content-negotiation, Long Polling, HTTP-2 etc) (qadeer to add)		- https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/modern-web-applications-characteristics
				Single Page Apps, http://singlepageappbook.com/goal.html
				Sockets - https://socket.io/docs/v4/
				SSR-CSR (Server side rendering vs client side rendering)
				 https://medium.com/walmartglobaltech/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff/2cefe8 https://www.toptal.com/front-end/client-side-vs-server-side-pre-rendering
				Caching - https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching
				Prowser Rendering - https://medium.com/@monica1109/how-does-web-browsers-work-c95ad628a509
	Soft Skills	How to use: - StackOverflow	0	- Ingus miediani.com gamonica i rosmow-oces-web-biomsets-monk-tsudauzzasus
	4.01	- Google		
	API	- Node.js, NPM, Yarn - Express.js with CRUD APIs	2	<raafay added="" to=""> CRUD API</raafay>
		- Middlewares - Pagination		https://www.codementor.io/@olatundegaruba/nodejs-restful-apis-in-10-minutes-q0sgsfhbd
		- Validation - Authentication and Authorization (JWT, 0Auth)		Nodejs Fundamentals: (essentials)
				https://frontendmasters.com/courses/node-js-v2/
				NodeJs API design fundamentals:
				https://frontendmasters.com/courses/api-design-nodejs-v3/
				Nodejs Advanced (express and mongo): (essentials) https://frontendmasters.com/courses/api-node-rest-graphql/
Backend Development		SQL Mongo	1	Database fundamentals: https://frontendmasters.com/courses/databases/
		mongo		(qadeer to add courses for DB queries)
	Cache	- Cache - Redis	0.5	(bilal)
	System Design	- Queues		
		System Design Patterns (MVC, SOA, AOP, Microservices) (bilal to add) Deployment on heroku	0.5	Intro to containers: (elective)
		Docker/Docker compose Deployment on EC2		https://frontendmasters.com/courses/complete-intro-containers/
	Code Quality	Ensuring Code Quality (such as Linter etc) Clean Code Principles		
Frontend Development	Foundation	- HTML - CSS (flexbox, css-grid, responsive design)	1	Web dev fundamentals: https://frontendmasters.com/courses/web-development/
		- CSS (nexcox, css-gnd, responsive design) - Browser Understanding (inspection, debugging, profiling etc) - API calling and Socket integration basics		
		- Ar i canning and SUCKET Integration basics		- HTML https://www.w@cchoole.com/html/default.gep
				https://www.w3schools.com/html/default.asp
				CSS https://www.w3schools.com/css/default.asp
				JQuery https://www.w3schools.com/jquery/default.asp
	React	 - VirtualDOM and React Lifecycle (class-based hooks) - Components / Structure / Hirearchy / Best practices / Patterns such as composition vs. inheritance 	2	React: https://frontendmasters.com/courses/complete-react-v6/
		- Local State Management - Event Handling		FreeCodeCamp Course:
		- Conditional Rendering		https://www.youtube.com/watch?v=4UZrsTqkcW4
		- Props - Pure vs. impure Functions - Hooks API		
		- Context API - Separation of Concerns (separate UI and business-logic)		
		Redux (hig topic!) HOCs (Higher Order Functions)		
	React Native	- Flatlists	1	React-native:
		- other topics		https://frontendmasters.com/courses/react-native-v2/ (Hunain to add resources for all topics)
		- Inter-application communication	0.5	(nam to add)
	System Design	- Micro-frontends - UI / Component Libraries / Design system		
Infrastructure	Command Line / Bash		0.5	https://github.com/jjevy/the-art-of-command-line
	Pipelines AWS		0.5	AMC in Diain Fastish, https://generalis.doco.uth.com/generalis.doc
	AWO		2	AWS in Plain English: https://expeditedsecurity.com/aws-in-plain-english/ AWS Exam: https://acloudguru.visme.co/view/31vvzvg1-s01-l02-exploring-the-exam-blueprint
				THE COURT IMPOST RESIDENCE AND ASSESSED THAT I THE TAXABLE PROPERTY OF THE TAXABLE PROPERTY.
		Total Time	15	

Topic or Area	Suggested By	Suggestion		
	Qadeer	Setup a task board (jira or trello) for each fresh graduate so that tasks are tracked, as well as start/end dates are recorded		
	Umair	We should expect one MR daily for a task		
	Hunain	Should have demo of projects/tasks after every 15 daysm in the EM call		