Technical Requirements
&
Learning Plan
COSMIC

Primary Instructor	**Anjana Shah**
Team Member	Angela Efremova
Team Member	Le An Nguyen
Team Member	Renzzi Adorador
Team Member	Ronak Gala

Document Revision History

Revision #	Date
0.1.0	11/23/2022
1.0.0	11/27/2022

Technological Requirements

	Technology	Pro's	Con's
Database	MongoDB	● Easy Environment and a Quick Set-up ● The installation, setup, and execution f are quick and simple. It is faster and ea than RDBMS and offers modern JavaS frameworks. ● This feature has allowed us to confider NoSQL structures. It also provides qui and training opportunities than SQL da ● MongoDB's schema is not predefined. it has a dynamic schematic architecture with non-structured data and storage. V value this feature mostly for the develo testing phases. ● Businesses keep evolving and so do the maintain. It is important to have a flexi model that could adapt to these change ● Sharding and Scalability are other featu might be useful during the developmer project	● MongoDB offers high-speed performar right indexes. In case the indexing is in incorrectly or has any discrepancies, M perform at a very low speed. Fixing the indexes would also consume time. This one of the major limitations of MongoI
Programming Languages	Javascript	● Speed and adaptability, as it can be use front end to the back end for a variety c embracing it, we can make application and development easier while also sim learning process to meet business need ● Server Load - operates on the client si validation is possible on the browser it can double-validate both on and off ser ● Versatility-capable of front-end as well development	● Lack of debugging resources ● Runs differently depending on the brov plan to deploy an web application versi business side.
Library	ReactJS	● ReactJS is much easier to learn and use with a good supply of documentation, t training resources that we can refer and ● It provides less coding and gives more It makes use of the JSX (JavaScript Ex which is a particular syntax letting HTI and HTML tag syntax render particular subcomponents. It also supports the bu machine-readable codes. ● A ReactJS web application is made up components, and each component has i controls. These components are respon outputting a small, reusable piece of HT which can be reused wherever you nee	● Has a high pace of development since t environment continually changes so fa of the developers are not feeling comfo relearning new ways of doing things re may be hard for them to adopt all these with continuous updates. ● Poor documentation is such a problem technologies update and accelerate so f is no time to make proper documentati ● ReactJS Covers only the UI Layers of t nothing else. So you still need to choos technologies to get a complete tooling development in the project.

	Axios	• Supports a wide variety of browsers in Chrome, Firefox, Edge, and Safari, and browsers such as IE11. So, it solves co related problems. • Allow canceling requests and request t • Be able to intercept HTTP requests. • Supports download progress for schedu reports download requests from the bus • Transforms the responded data to JSON automatically.	• Must be installed manually and import does not come along with Node.js.
Hardware		• As we use Azure (mentioned below), a physical hardware is not necessary.	• Highly dependent on a third-party side project.
Software	ExpressJS	• The Express.js framework enables us t JavaScript as your programming langu the front and back ends. It also allows to become full-stack developers. As a r development process is much faster an because we can be responsible for both presentation and data access layers. • Express.js is one of the most popular N frameworks, with an active open-sourc that constantly reviews and improves th • Ease the Rest API with the Express bui since it uses JavaScript as its sole prog language. Given that MongoDB is Java oriented, its use also plays a significant system will be including ○ Client Side ○ API server ○ Database Besides, ReactNative and ReactJS are Javascript-based, so using ExpressJS w variety of tasks for my team. • Suit all of the purposes for this project on server-side, multiple-page, hybrid w applications • ExpressJS eases the learning since Exp well-documented and supported by a la community.	• Node.js is a single-threaded framework event loop that listens to various events executes registered callbacks. So, there event stuck in a callback loop. Hopeful version of Node.js supports the use of which will increase the code quality an problems with callbacks.

	React Native	• App development speed and cost – to r and recycle components developed bef also handle app development for both p iOS and Android. • Be able to build high-performance app absolutely great user experience. • React Native community is massive, it save time in the testing and debugging we could find the solutions that we mig • The only maintenance task is to deal w codebase.	• Performance is still lower than native b Native does not allow the use of all the and potential of a specific platform. • Not efficient for complex interfaces – I consider complicated designs or advan interactions as a crucial part of your bu advantage, you should definitely go for development. • Lack of some custom modules – many modules are already available, yet, ther need for some specific components tha have to build from scratch ourselves. • Updating issues – it is hard to keep the with the latest React Native version. U Native versions is in most cases a comp process.
Runtime Environment	Node.js	• For compatibility, Node.js is a cross-pl solution which is used as a Javascript r environment as the other parts of this p as ExpressJS, ReactJS, React Native, a all Javascript based. • Node.js helps us to fill the big gap betw browser and application layer when co Javascript (Node) with other languages both browser and server in JavaScript. • Node.js can handle a large number of s connections, so it will be useful for ser applications interacting with third-party and retrieving data from them. • Node.js is a great choice for building th of applications for tracking website vis visualizing the outcoming data. So, wh investigate our traffic as soon as they a track of all of the end-devices and anal • It is also capable of building applicatio updated in real-time. Node.js is also an and non-blocking-driven server that ca large number of real-time users.	• Poor handling of high CPU activities s heavy block of requests demanding hig computations despite its excellence in I input-output process. • Since Node.js is capable of running on even without a web browser. It not only rules of types management, but it migh perform implicit type casting

Hosting Serv	Azure	● As it serves as a PaaS, all we need fron pipeline, storage comes with Azure out after setting up. ● Flexible plans which is based on the us traffic so we can scale up the hardware needed. There are a vast number of cor so we can choose to up/downgrade to s budget. ● Azure cloud is one of the most reliable infrastructure. The Azure Service Leve (SLA) cites an uptime of 99.95%, whic a little over four hours yearly of downt ● Comes along with administrative tools backup and recovery, and big data anal ● Security from cloud platforms in gener the traditional on-premise system since security features are always up-to-date.	● There isnt any recorded failure from Az But being highly dependent on an "All platform" itself has already been a risk the solution is to backup our system fre constrain the loss of data when there is happening.

Learning Plan

Team member Roles and Tasks distribution

No.	Team Memb	Task(s)	Current Skills level*	Start date	End date	Note
1	Angela Efrem	• Front-end development • UX/UI	Figma [:::::::: 80	2022-11-08	2022-11-24	Sources: Youtube videos
			React-Native [:::: 50%	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			ReactJS [::::: 50%	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			NodeJS [::::::: 70	2022-10-26	2022-12-21	Source: Youtube videos
			JavaScript [::::::: 70	2022-10-04	2022-12-21	Source: Youtube videos
2.	Le An Nguyer	• Full-Stack Business-Side Application • Web Server Development	Devops Development [::::: 60%	2022-11-	2022-12-21	Sources: - Class lectures - Youtube videos
			Axios [::::: 60%	2022-11-04	2022-11-29	Sources: - Class lectures - Youtube videos
			ExpressJS [::::: 50%	2022-11-	2022-11-30	Source: Youtube videos
			React-Native [::: 40%	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			ReactJS [::::: 50%	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			NodeJS [::::: 50%	2022-10-26	2022-12-21	Source: Youtube videos
			JavaScript [::::::: 70	2022-10-04	2022-12-21	Source: Youtube videos
3.	Renzzi Adorac	• Front-end development • UX/UI development	Figma [:::::::: 80	2022-11-08	2022-11-24	Source: Youtube videos
			React-Native [::::: 60%	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			ReactJS [::::::: 70	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			NodeJS [:::::: 80%	2022-10-26	2022-12-21	Source: Youtube videos
			JavaScript [:::::::: 80	2022-10-04	2022-12-21	Source: Youtube videos

4.	Ronak Gala	• Full-Stack M Application Development	Devops Development	[::::::: 70	2022-11-02	2022-12-23	Sources: - Class lectures - Youtube videos
			Axios	[::::::: 70	2022-11-11	2022-11-29	Sources: - Class lectures - Youtube videos
			React-Native	[::::::: 70	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			ReactJS	[::::::: 70	2022-11-07	2023-01-07	Sources: - Class lectures - Youtube videos
			NodeJS	[::::::: 70	2022-10-26	2022-12-21	Source: Youtube videos
			JavaScript	[:::::::: 80	2022-10-04	2022-12-21	Source: Youtube videos

** Indicated levels are relative and subjective to the matter of time when this document was written*