

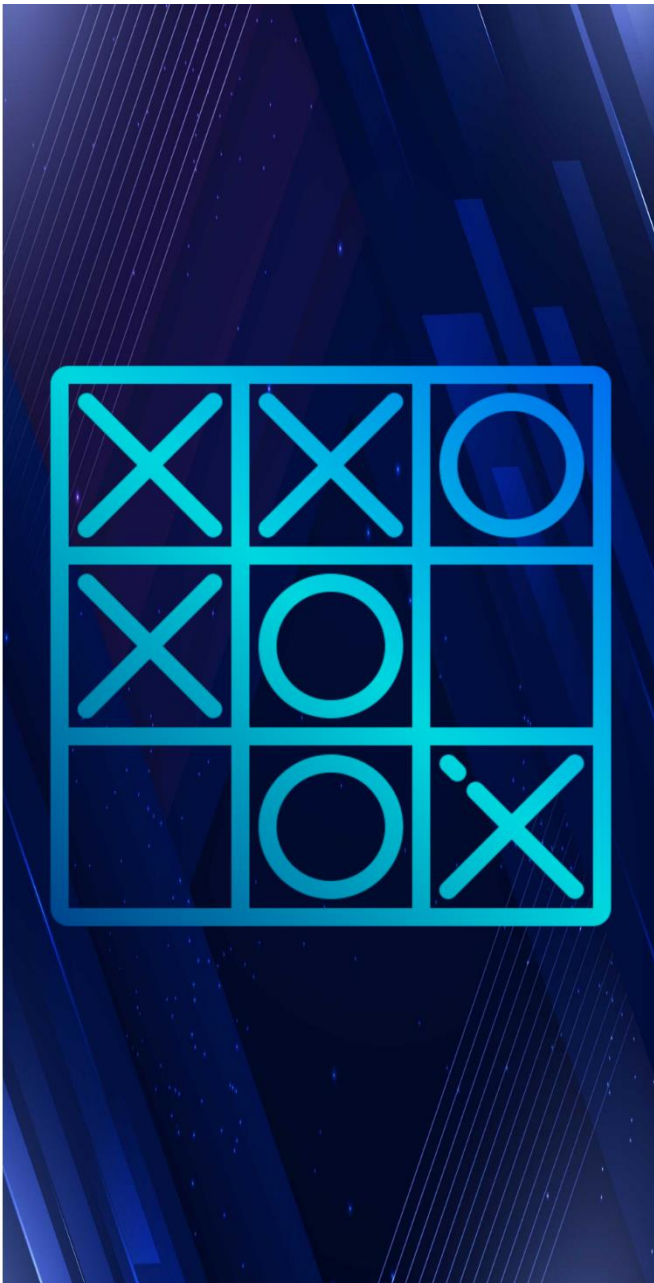
SWE Challenge- THE MINI PROJECT (Tic Tac Toe Online)

Description: Build an Android mobile app developed with front-end React Native, backend

Serverless architecture for a multiplayer online tic-tac-toe mobile app.

Date: 10 July 2020

Author: Shermaine Lim Si Hui



Contents

SWE Challenge- THE MINI PROJECT (Tic Tac Toe Online)	1
Technologies used	3
Extensions used in Visual Studio Code	3
Development.....	3
1a. Instructions on how to load the application, create a game session and play the game on 2 or more phones	3
Testing APK on PC	3
Testing APK on Android Phone	7
1b. API specifications to your production APIs, with sample API requests in Postman	8
1c. System Architecture	9
Flow of the application.....	9
Firebase SDK, Firestore Database & Cloud Functions	10
Security	11
What is a Serverless Architecture?	12
AWS Lambda versus Firebase: Which to Choose?	12
Ease of deployment	13
1c. Infra choices	13
Future Improvements: Garbage Collection	13
The benefits of Garbage Collection in Cloud	14
1d. Any assumptions made with regards to the take home assignment	14
Learning Journey	14
What is TypeScript?	14
Why use TypeScript over JavaScript?	15
What is Prettier extension in VS Code?	15
What is the difference between ReactJS and React Native?	15
What is the difference versus APK versus app-bundle?	16
What is node.js used for with React Native?	16
What is Expo.io used for together with React Native?	16
What is the difference between const and let?.....	16
What is the difference between ReactDOM.render and Rendering Elements?	17
What are props and components	17
Props versus States	17
What is async and await	17
Why use NOSQL for this project? NOSQL(DBMS) versus SQL(RDMS)	17
Why use === and not double ==?	18
What is the difference between ts and tsx?	18
What is the difference between class and function component with Hooks?	18
Function Component with hooks.....	19

What is the difference between Yarn and npm?	19
What is a package manager?	19
What is the difference between useState & useEffect?	19
What is object destructuring?.....	20
What is the difference js and jsx?	20
What is tsx?	21
What is babel.config.js?	21
What is tsconfig.json?	21
References	21

Technologies used

Visual Studio Code, Expo.io, Node.js, Android Studio Emulator, Firebase, Firestore, Cloud Functions (Serverless Architecture), BlueStacks, MEmu, Android Phones.

Extensions used in Visual Studio Code

Prettier (For structuring codes to be neater)

Development

- Developed with React Native and Expo.io to build Android APK
- Written with TypeScript.
- Configured with NoSQL database

1a. Instructions on how to load the application, create a game session and play the game on 2 or more phones

To test the android game application (APK) on PC, we can use BlueStacks and MEmu which are Android emulators to allow users to play android games and apps on PC.

To test the android game application (APK) on an Android Phone itself, download the APK file, press and install it.

Both steps achieve the same results when testing for Android.

Testing APK on PC

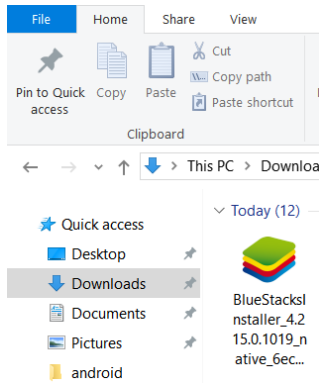
- 1) Install BlueStacks and MEmu to simulate 2 phones and play the game. Links for installation provided.

BlueStacks: <https://www.bluestacks.com/>

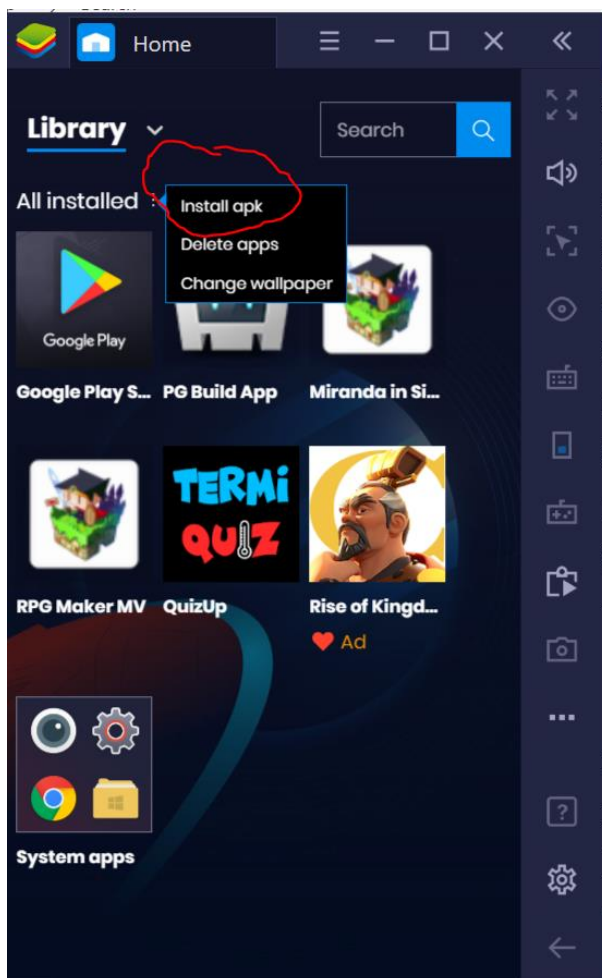
MEmu: <https://www.memuplay.com/>

2) Instructions how to install the apk file on mobile to test:

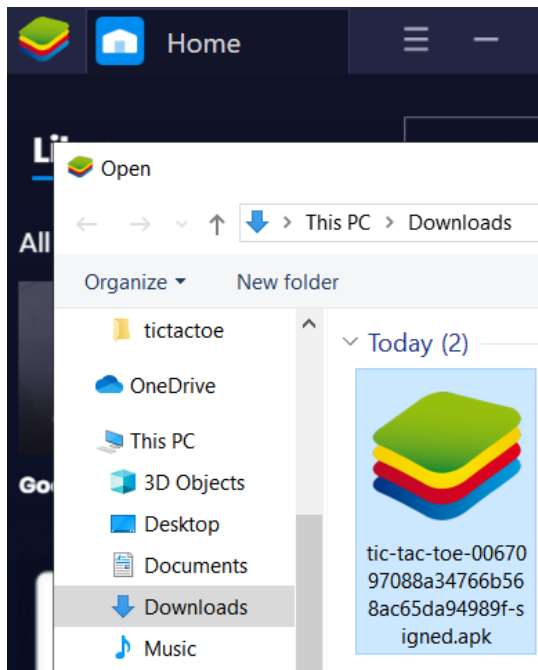
After using Expo.io to build the Android APK file and downloading it, locate where you downloaded and stored the APK file. Example of how the APK file will look like when you download the finished APK build from Expo.io.



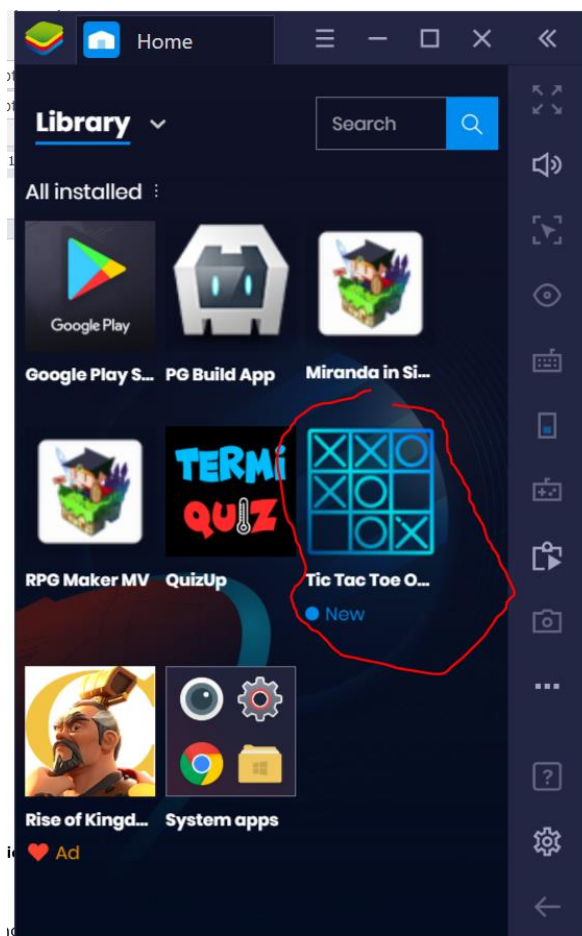
3) Now open both BlueStacks and MEmu. For BlueStacks, navigate to Library and press the button for Install apk circled in red.



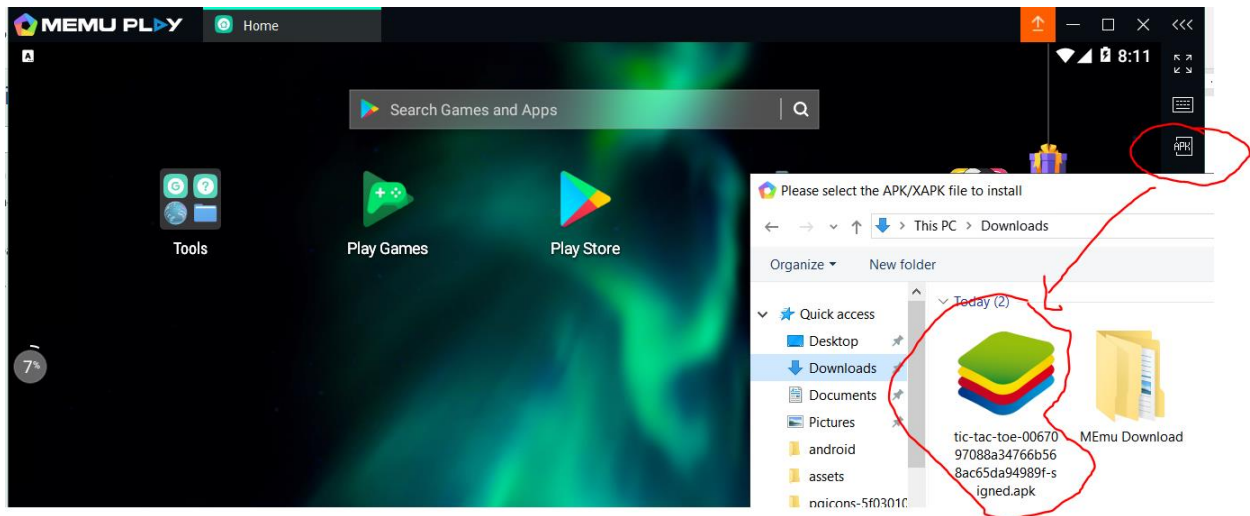
- 4) It will prompt you to the file explorer and click on the APK file to install.



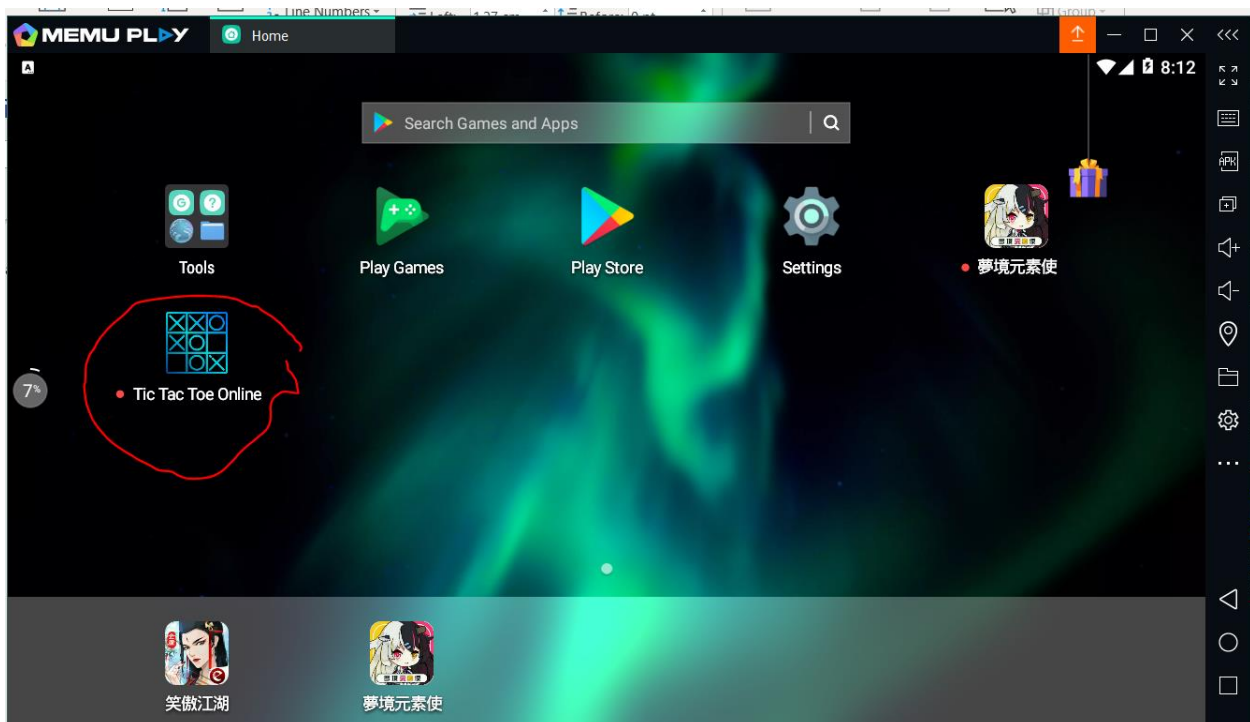
- 5) The Tic Tac Toe Online will be installed as circled in red.



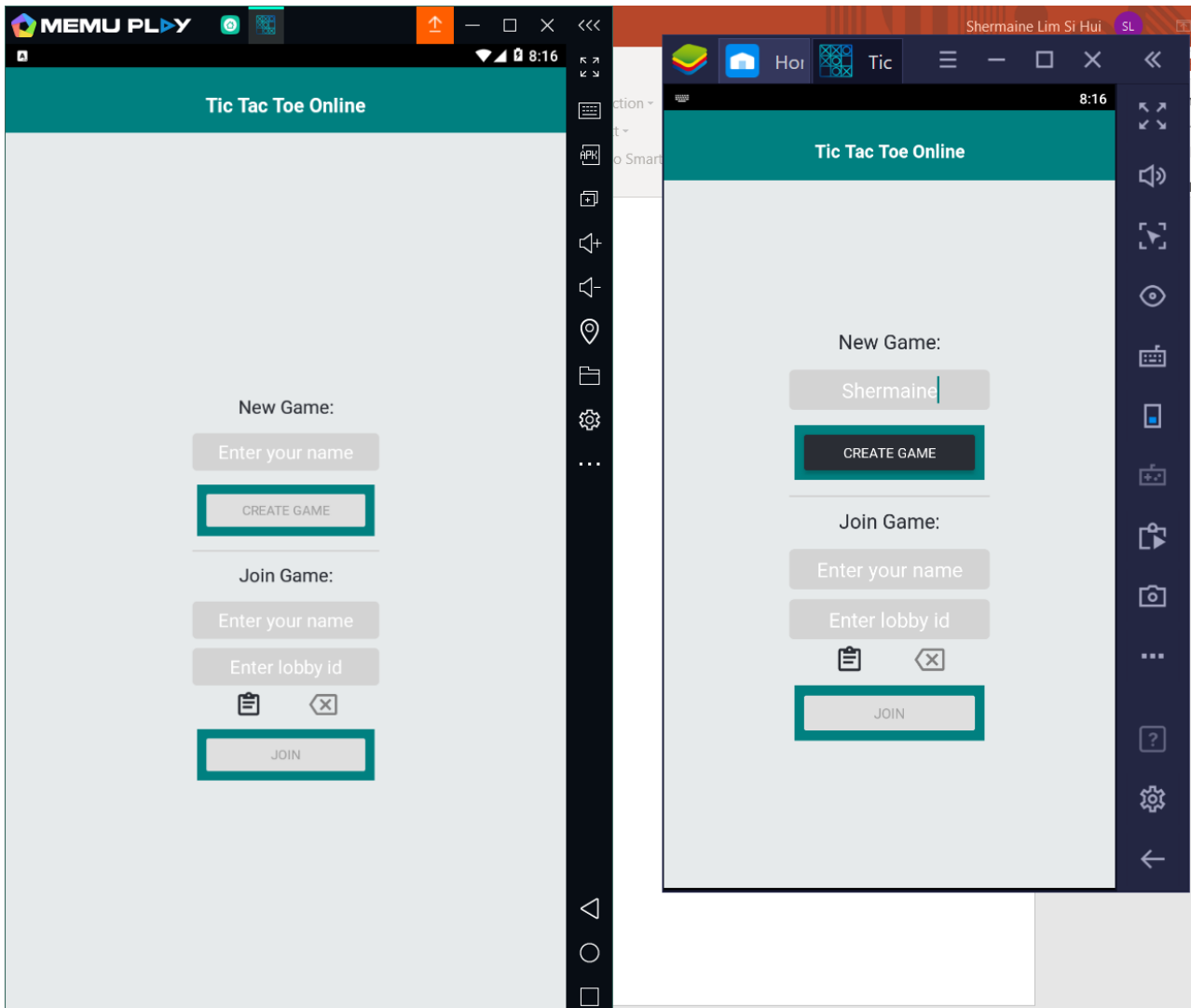
- 6) For MEMu, looking at the red circles, click on the APK icon at the right corner and it will prompt open the file explorer. Install the APK file.



- 7) Tic Tac Toe Online has been installed on MEMu as circled in red.



- 8) On both BlueStacks and MEmu, click on the Tic Tac Toe Online icon game and it should load giving you the following screens.



- 9) Please refer to video walkthrough on how to create a game session and play as 2 players online and the data reflected in the Cloud database.

Private Video Link on YouTube: <https://youtu.be/NEfy2yENzxc>

Testing APK on Android Phone

- 1) Obtain 2 Android Phone to test the game. Download the APK file from the link provided below.

Google Drive Link to download the APK file to Android Phones:

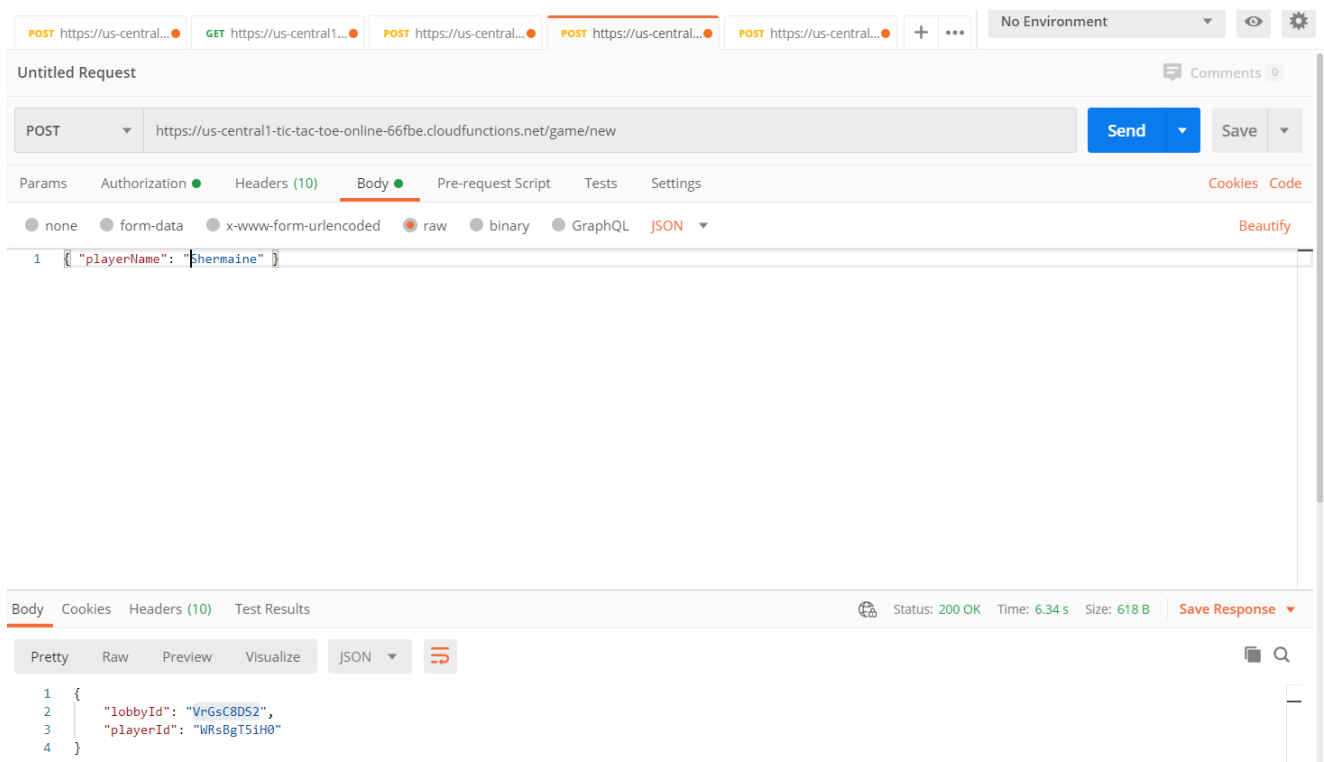
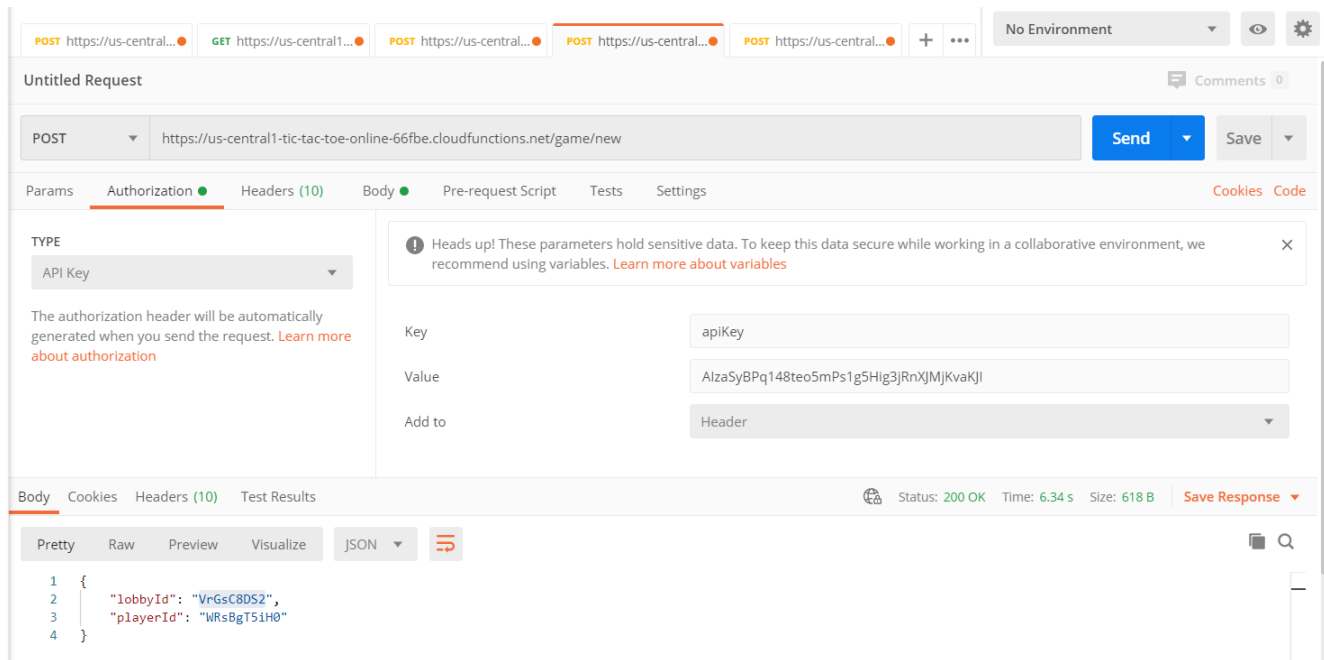
<https://drive.google.com/drive/folders/1VT2Pltari5BucA8ux1QjgKKwRFeF-0mx?usp=sharing>

- 2) Please refer to video walkthrough on how install and open the APK file on Android Phone.

Private Video Link on YouTube: <https://youtu.be/iuCTRArwgLM>

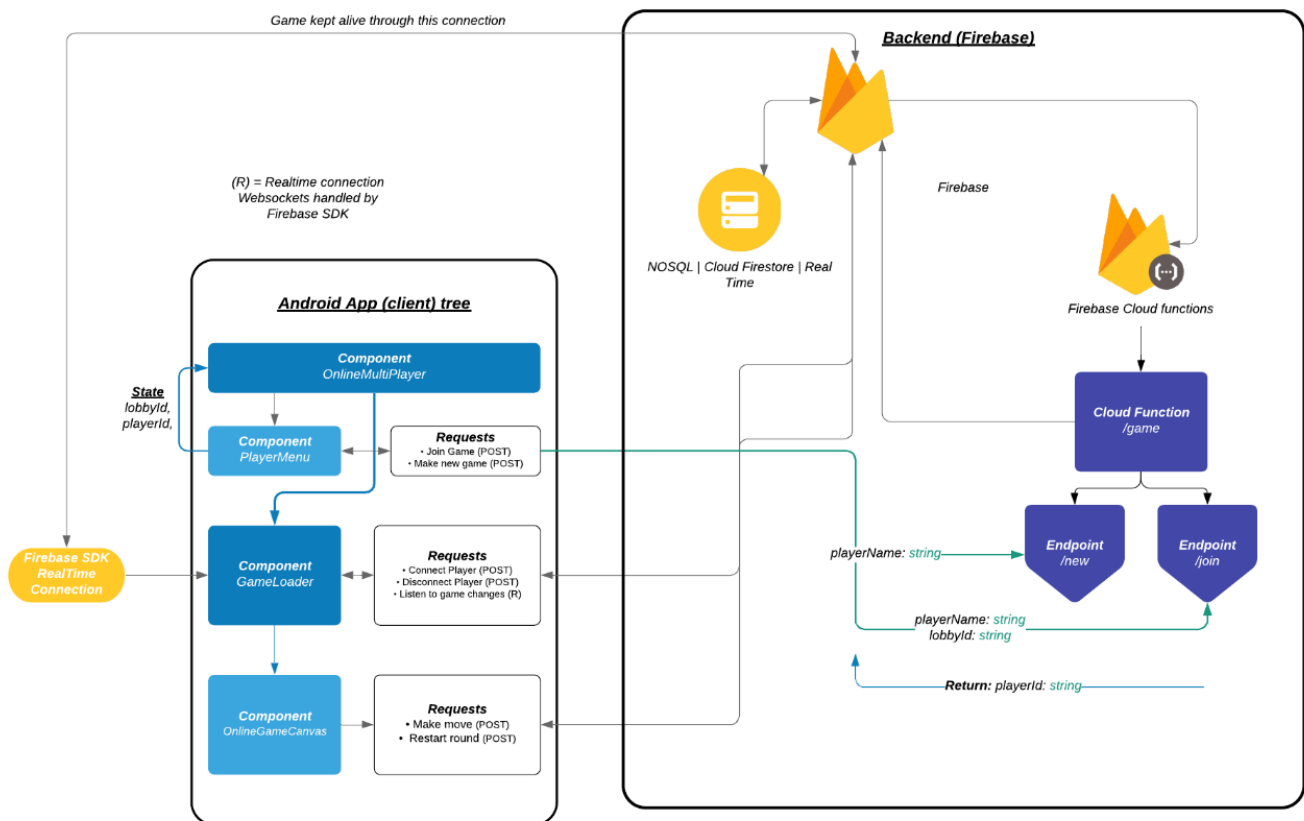
1b. API specifications to your production APIs, with sample API requests in Postman

Firebase Cloud Functions API. Using Postman, use POST request, enter the link, add /new behind, in Authorization, proving apiKey and value. Add the playerName and name. E.g. `{"playerName": "Shermaine"}`.



1c. System Architecture

System architecture of Multiplayer Online Tic Tac Toe Android Game



Flow of the application

The system architecture of the Multiplayer Online Tic Tac Toe Android game consists of the Front-End and Back-End. The frontend (left) is the Android App (client) tree while the backend (right) sits on Google Firebase. For the client side, it starts with the OnlineMultiPlayer Component, this is the main component of the app, wrapper and where most of the logic sits.

The Online MultiPlayer component will then render two other components. The PlayerMenu is supposed to make a POST Request to the /new Endpoint when the player wants to -Make a new game or make a POST Request to the /join Endpoint when the player wants to -Join Game. The PlayerMenu which has a few tasks/requests where the method of the requests is POST.

For the green arrows going to the different Endpoints, the different data types are set so the parameters that need to be parsed with the request, so for the /new game endpoint, playerName is needed whereas for /join game endpoint playerName and the lobbyId is required.

Upon that, the playerId is also sent back, which is stated by the return blue arrow. Therefore, when that data is sent back to the PlayerMenu component which then passes the lobbyId and playerId states into the OnlineMultiPlayer Component.

Once it gets there, it will render the Game Loader component which has the tasks/requests that it needs to Connect Player, Disconnect Player and Listen to game changes and those arrows will point towards the Cloud Firestore. This means that Firebase SDK will be utilized to make those calls, and it is not going to talk to any Cloud Function endpoint, but is going to talk directly to the Firestore Database through the Firebase SDK.

It also needs to also make a real time connection to listen to any game changes, and this component collects all that data and passes the data down to the OnlineGameCanvas component. The OnlineGameCanvas component renders the whole game canvas and needs to make 2 requests. Whenever players presses on one of the tiles to choose, the make move POST request is sent out, and whenever user presses on the new game, it will restart the round, sending a Restart round POST request to the Firebase SDK and that will then update the database.

That sums up the entire flow of the application.

Firebase SDK, Firestore Database & Cloud Functions

Firebase SDK sits in the middle, is in the app and it is in the server side. As a separate layer, Firebase provides the functionality to communicate directly with the database directly so there is no need to go through a server like Firebase Cloud functions to do that. There can just be the client to database connection through the Firebase SDK.

The Firebase SDK handles both real time and one-time request, such as get and post request. The Firebase SDK is preconfigured from Firebase, so it needs to be configured into the app to provide the connection string which is the Firebase configuration to privatised in the environment variable.

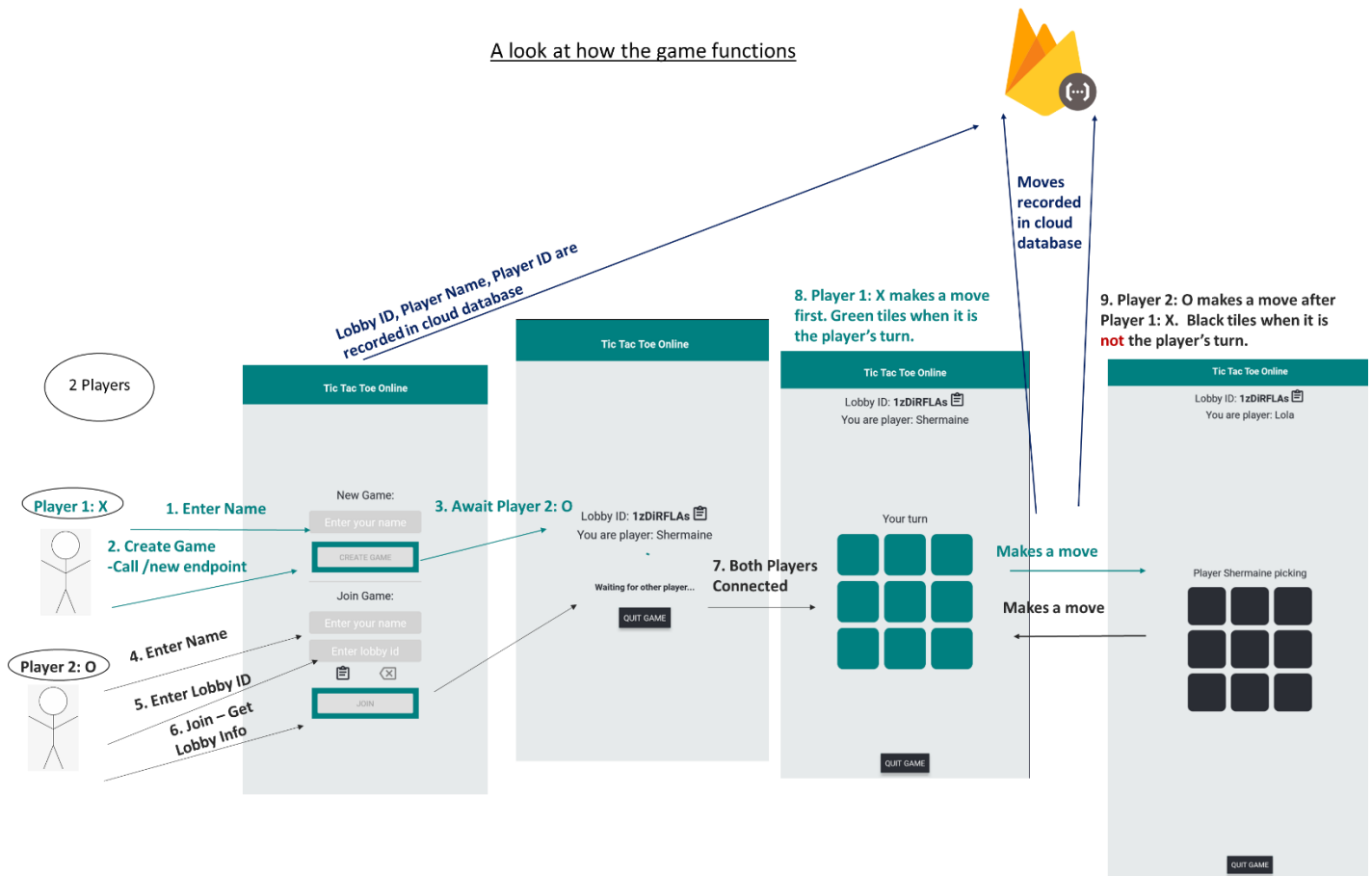
The apiKey supplied to the client so a connection can be established to the Firestore Database through the Firebase SDK. Therefore, necessary configurations need to be made at Firebase first to get the apiKey. Firebase is the main wrapper for the backend and related to Firebase Cloud Functions. Cloud functions is a service on Firebase that is used and configured into the files to be able to deploy a node.js function.

Then the Cloud Functions spins up a node.js server which needs to warm up and stays up for at least a few mins, so whenever a user makes a request, that function stays warm and then after a few mins, it goes inactive, killing the server again, which makes it very efficient and cost effective.

Firebase supplies the API for the Cloud Functions and the Cloud Functions can talk to the Firestore database and it does that through the Firebase SDK which is on the client side.

Security

Cloud Functions is used to initialize the creation of the lobby, to prevent the user from going to the client to create a bunch of lobbies and utilizing bots or launching attacks like Denial of Service (DoS), or other forms of attacks that will limit the data request because Firebase has limits for requests. Therefore, by using an endpoint on the cloud function, this allows only the function to create the game and not the client which is a security feature.



A look at the data records in Cloud Database



Various Lobby ID , date/time created, and fieldTypes of Tic Tac Toe Grid with moves stored

tic-tac-toe-online-66f6e	lobbies	1zDIRFLAs
+ Start collection	+ Add document	+ Start collection
lobbies	1zDIRFLAs	+ Add field
	4UIZSD7EJ	createdAt: July 9, 2020 at 6:49:00 PM UTC+8
	F_N4vC-Sd	fieldTypes
	m9Y33aEsr	0 'x'
		1 'x'
		2 'x'
		3 'x'
		4 'x'
		5 'x'
		6 'x'
		7 'x'
		8 'x'
		gameStarted: true

Player Name and ID are stored.

tic-tac-toe-online-66f6e	lobbies	1zDIRFLAs
+ Start collection	+ Add document	+ Start collection
lobbies	1zDIRFLAs	+ Add field
	4UIZSD7EJ	players
	F_N4vC-Sd	0
	m9Y33aEsr	connected: true
		id: '9sXgyQJZEa'
		name: 'Shermaine'
		1
		connected: true
		id: 'uzwcvP5gqj'
		name: 'Lola'
		resetable: false
		startingPlayer: '9sXgyQJZEa'
		turn: 'uzwcvP5gqj'

What is a Serverless Architecture?

A serverless architecture is essentially a microservice architecture. Whenever an event occurs, a code or function is executed. The server runs when something happens. The code only wakes up when it receives a request. It's not persistent and running all day, all week.

It only uses resources when it needs to. The benefit of this is that each function has its own responsibility, and none of it overlaps, making things cleaner and less finicky.

AWS Lambda versus Firebase: Which to Choose?

AWS Lambda's performance is very strong. The service is reliable and customer support is superb. It's also a pay-per-use service so that your cost is dependent on the type of traffic you receive. In general, their entire ecosystem is made to be all-inclusive. If you don't want to bounce around with services, AWS is a great choice. AWS has a slight learning curve, especially considering they have such a large suite of products on hand. Their infrastructure is also geared a little more towards larger teams such as enterprises. If handled incorrectly, AWS could end up costing you more money than you originally anticipated.

Google's Firebase includes Cloud Functions and services that make it easy to scale. It also features traffic management services within their suite. They also boast low IT costs. If you're a small team setting up, Firebase might serve you better than a heavy AWS setup. It also has a lower learning curve than AWS.

(AWS vs Firebase - Is It Even a Fair Fight? 2018).

Ease of deployment

Ease of deployment is one of the most important functionalities because easy deployment means how fast you can deliver the product to your users.

AWS Lambda	Google Cloud Function
<ol style="list-style-type: none">1. Fill basic info2. Add API gateway3. Add trigger4. Write function code	<ol style="list-style-type: none">1. Fill basic info and write function code

List of step comparison

From the table above, there are a lot of steps on AWS Lambda while Google Cloud Function has only one step to deploy the function. It is because AWS Lambda provides many options which the users can config their function.

1c. Infra choices

AWS Lambda and Google Cloud Function both have efficient scalability AWS Lambda works properly with the users who want flexibility and more freedom to setting their own functions. While Google Cloud Function works properly with the users who want simplicity in their work and easier deployment.

Both AWS Lambda and Firebase Cloud Function are serverless architecture. Firebase Cloud Function was used for the Tic Tac Toe application for its simplicity, ease of deployment, efficient scalability and low cost compared to a heavy AWS setup.

Future Improvements: Garbage Collection

As for future improvements, to ensure the Tic Tac Toe mobile application delivers optimal performance and does not accumulate so many data of obsolete lobby IDs and other details from many users , garbage collection can be implemented to clean the cloud database periodically to keep costs down by clearing storage of expired and obsolete data as the accumulation of data will incur higher costs.

The benefits of Garbage Collection in Cloud

Garbage collection is the automatic, ongoing process of removing expired and obsolete data from Cloud Bigtable tables. A garbage-collection policy is a set of rules you create that state when data in a specific column family is no longer needed.

Garbage collection is a built-in, asynchronous background process. It can take up to a week before data that is eligible for garbage collection is removed. Garbage collection occurs on a fixed schedule that does not vary based on how much data needs to be removed. Until the data is removed, it will appear in read results. You can filter your reads to exclude this data.

The benefits of garbage-collection policies include the following:

- Minimize row size - You always want to prevent rows from growing indefinitely. Large rows negatively affect performance. Ideally, you should never let a row grow beyond 100 MB in size, and the limit is 256 MB. If you don't need to keep old data, or old versions of your current data, using garbage collection can help you minimize the size of each row.
- Keep costs down - Garbage collection ensures that you don't pay to store data that is no longer required or used. You are charged for storage of expired or obsolete data until compaction occurs and garbage-collected data is removed. This process typically takes a few days but might take up to a week.

(Google Cloud, 2020)

1d. Any assumptions made with regards to the take home assignment

- There are 2 players playing Tic Tac Toe Grid using an Android Application.
- They will be playing it more than 1 time in a single session.
- Player names are recorded in the Firestore database.

Learning Journey

I was new to these technology and language so I had to read up before building. These were the questions that crossed my mind while I was learning how to build this project with the new technology.

What is TypeScript?

TypeScript is an extension of JS, is not something in React, TypeScript runs on VS Code.

Why use TypeScript over JavaScript?

TypeScript is an extension of JS, is not something in React. TypeScript gets compiled to JavaScript. Typescript gives static features.

TypeScript reduces error for parsing, reduces bugs by a lot, basically by integrating documentations in your code, the props, state or app. It is very helpful (avoid bugs and save time)

Advantages of using TypeScript over JavaScript

- TypeScript always point out the compilation errors at the time of development only. Because of this at the run-time the chance of getting errors are very less whereas JavaScript is an interpreted language.
- TypeScript has a feature which is strongly typed or supports static typing. That means Static typing allows for checking type correctness at compile time. This is not available in JavaScript.

(GeeksforGeeks ,2018).

What is Prettier extension in VS Code?

Prettier extension is a code formatter that saves a ton of time and makes your code more consistent such as formatting semi colon, the tabs, the spaces. it is a good practice to use those.

What is the difference between ReactJS and React Native?

React-Native is a framework, where ReactJS is a JavaScript front end library you can use for your website. When you start a new project with ReactJS, you probably will choose a bundler like Webpack and try to figure out which bundling modules you need for your project. React-Native comes with everything you need and you most likely wouldn't need more. When you start a new project you will notice how easy is it to set up: it's very fast and only takes you one command line to run in the terminal and you will be ready to go (What are the main differences between ReactJS and React-Native? ,2019).

React JS is front end library developed by Facebook. It's used for handling view layer for web and mobile apps. ReactJS allows us to create reusable UI components. It is currently one of the most popular JavaScript libraries and it has strong foundation and large community behind it.

React Native is a mobile framework that compiles to native app components, allowing you to build native mobile applications, iOS, Android, and Windows, in JavaScript that allows you to use React JS to build your components, and implements ReactJS under the hood (GoodWorkLabs, 2017).

React JS uses react-dom implementation to render the components to the DOM. This library is way larger than React JS itself. React Native has their own implementation of rendering components for native platforms.

- Both have their own rendering mechanism

- React Native has a flexbox way of styling instead of regular CSS for the web
- React Native must be compiled to native code where ReactJS can be used directly in the DOM

In conclusion ReactJS cannot use for native mobile development only for web mobile. React Native is for pure native mobile development.

What is the difference versus APK versus app-bundle?

APK (Android application Package) is the packaging format which eventually will be installed on device.

An Android App Bundle is a publishing format that includes all your app's compiled code and resources and defers APK generation and signing to Google Play.

Google Play uses your app bundle to generate and serve optimized APKs for each device configuration, so only the code and resources that are needed for a specific device are downloaded to run your app. You no longer have to build, sign, and manage multiple APKs to optimize support for different devices, and users get smaller, more-optimized downloads (Android Developers, 2020).

What is node.js used for with React Native?

React Native uses Node.js, a JavaScript runtime, to build your JavaScript code. Node.js is a server-side JavaScript runtime environment. React Native ships with some tools that are written for Node.js.

Node.js is an open source platform built on Chrome's JavaScript runtime; it offers a way to easily build fast, scalable programs. Node.js allows you to run JavaScript in Terminal and helps create modules.

Download node.js from nodejs.org. This JavaScript runtime gives you access to npm, which is a convenient tool created by the node.js project that you can use to manage open source packages. Make sure that you download the latest LTS (Long Term Support) version of node.js. Also included with this download is a development server called the Metro bundler, which provides live updates when debugging. (W. and Wilms, J. ,2019)

What is Expo.io used for together with React Native?

Expo is a framework and a platform for universal React applications. It is a set of tools and services built around React Native and native platforms that help you develop, build, deploy, and quickly iterate on iOS, Android, and web apps from the same JavaScript/TypeScript codebase (Expo Documentation ,2020).

What is the difference between const and let?

Const is almost the same as let. However, the only difference is that once you've assigned a value to variable using const, you can't reassign it to a new value. Variables declared with let can be re-assigned, but variables declared with const can't be. However, just because a variable is declared with const doesn't mean it's immutable, all it means is that the value can't be re-assigned (UI.dev ,2020).

What is the difference between ReactDOM.render and Rendering Elements?

ReactDOM renders React Elements to the DOM, so it's for web development. ReactNative renders React Elements to a native mobile platform (android or iOS). You can write your code to target both at the same time.

What are props and components

Most components can be customized when they are created, with different parameters. These created parameters are called props, short for properties.

For example, one basic React Native component is the Image. When you create an image, you can use a prop named source to control what image it shows. This is how you can reuse components. You can make a universal component and pass it different props to customize it to the use case (React Native, 2020).

Props versus States

Props is what u pass to the component, but state is how we control the data flow in the react component. We can update the state in the React Component to re-render it. This means we are showing the updated data.

What is async and await

An async function returns promises that are resolved with function's return value or rejected with uncaught errors. The “await” keyword tells the program to temporarily exit the async function and resume running it when a given task completes (React Native Meets Async Functions ,2016).

Why use NOSQL for this project? NOSQL(DBMS) versus SQL(RDMS)

There is flexibility in the database structure (DBMS) over (RDMS). There is no need for the relational database for this use case because it is simple, and we just store all our values in the record. It integrates well with JavaScript because it uses JSON. JSON is a JavaScript object notation, this means that instead of converting what we get from the SQL RDBMS, we can take what we have from the NOSQL and use it. Basically, the data structure in NOSQL is more suitable for our application.

JSON is a dictionary structure (key and value).

Why use === and not double ==?

JS can use triple === and double ==, === checks for type too.

The triple === prevents the type from converting to other types. e.g. from numbers to string.

What is the difference between ts and tsx?

The ts will not return any tsx (the green stuff(components)), if u return any tsx, it needs to be named .tsx. ts is used for normal js without react.

TSX allows for embedding JSX elements inside the file and is largely used by React. TS files are plain old Typescript files and do not support adding JSX Elements. Typically, you would separate concerns by placing only code relating to presentation in the TSX files (A., & J, 2019).

What is the difference between class and function component with Hooks?

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class. The idea behind introducing Hooks is to help reduce the code that one must write and aggregate similar actions together. There are a few good reasons to make use of Hooks instead of classes

It's hard to reuse stateful logic between components Generally when you use HOC(High Order Components: rendering components in another components) or renderProps you have to restructure your App with multiple hierarchies when you try to see it in DevTools, Hooks avoid such scenarios and help in clearer code.

Complex components become hard to understand Often with classes Mutually unrelated code often ends up together or related code tends to be split apart; it becomes more and more difficult to maintain. An example of such a case is event listeners, where you add listeners in componentDidMount and remove them in componentWillUnmount . Hooks let you combine these two

Classes confuse both people and machines with classes you need to understand binding and the context in which functions are called, which often becomes confusion.

The idea is to be able write the code that you can write using React class component using function component with the help of Hooks and other utilities. Hooks can cover all use cases for classes while providing more flexibility in extracting, testing, and reusing code (Components et al., 2018).

Function Component with hooks

Hooks are a new addition in React 16.8. The most useful feature of Hooks is that it allows using state without using class.

There are two most commonly used hooks: the state hook -- `useState` and the effect hook -- `useEffect`.

State hook allows you to add states in the function component. Instead of setting an initial state with this. state statement in the constructor, we can import `{useState}` from `react`, which will allow you to set the initial state as an argument. State hook will return a pair of values: the current state and a function that updates it. Usually, we will use `useState` like this:

```
const [time, setTime] = useState(new Date())
```

Effect hook will get invoked with the first DOM updating. We can pass in a function in `useEffect`, and every time the DOM gets updated, the function in `useEffect` will get invoked too. Also, the effect hook allows you to pass in an array as the second argument, which contains all the dependencies that will trigger the effect hook. If any of the dependencies changed, the effect hook will run again. This feature provides us a more efficient way to make an Ajax request. Instead of making the request every time with DOM updates, you can pass in dependencies that only make the request while these values change **(DEV,2020)**.

What is the difference between Yarn and npm?

Yarn is a version of npm. Yarn is a package manager for react. It does the same thing.

Yarn is a new package manager that replaces the existing workflow for the npm client or other package managers while remaining compatible with the npm registry. It has the same feature set as existing workflows while operating faster, more securely, and more reliably (Nakazawa et al., 2016). (YARN vs. NPM ,2019).

What is a package manager?

It only installs the modules and libraries specified. Just share the `package.json` file and can `npm i` to download the whole modules.

What is the difference between `useState` & `useEffect`?

By default, `useEffect` runs after every render, including the first one. React guarantees the DOM has been updated by the time it runs the effects.

When you use `useEffect`, you're telling React that your component needs to do something after rendering. You pass a function (the "effect"), and React will remember it and call it later, after performing the DOM updates.

```
function MyComponent() {
```

```

React.useEffect(() => {
  // this side effect will run after every render
});
return ...
}

```

Both `useState` and `useEffect` enhance functional components to make them do things that classes can but functional components (without hooks) cannot:

- `useState` allows functional components to have state, like `this.state` in class components.

`useEffect` allows functional components to have lifecycle methods (such as `componentDidMount`, `componentDidUpdate` and `componentWillUnmount`) in one single API. (W. et al. ,2018)

`useEffect` takes a function which can contain any kind of operation including side effects. Any kind of side effect is not allowed inside the render method (or inside the function in case of a functional component). So, any kind of side effect should be used inside `useEffect`

`useEffect` also takes a second argument as an array [], in this array you can pass variables. When any of this variable updates it will cause the `useEffect` to run again, because we passed an empty array our `useEffect` will only run once throughout the life of our component.

If we didn't pass the second argument something like this

```

useEffect(() => {
  // hit an API
})

```

This will cause `useEffect` to run after every render, just like `componentDidUpdate` but I doubt you would be using this a lot because most of the time we don't want the side effects to run after every render (`useState` and `useEffect` explained ,2019).

What is object destructuring?

Destructuring was introduced in ES6. It's a JavaScript feature that allows us to extract multiple pieces of data from an array or object and assign them to their own variables. Destructuring really shines in React apps, where it can greatly simplify how you write props (Destructuring Props in React , 2018).

What is the difference js and jsx?

- Js is JavaScript, JSX is JavaScript React.
- Jsx is what you return for the components.
- .Js can return jsx and .jsx can return jsx
- .Ts cannot return jsx.

- Ts(TypeScript)
- .Tsx can return jsx

What is tsx?

- TypeScript React

What is babel.config.js?

babel.config.js is to support older browsers for newer version of JavaScript

What is tsconfig.json?

tsconfig.json is configuration for TypeScript, tells the TypeScript compiler, the preferences, if you want JavaScript etc. It exists for the TypeScript compiler.

References

(AWS vs Firebase - Is It Even a Fair Fight? 2018). Available at: <https://dashbird.io/blog/aws-lambda-vs-firebase/> (Accessed: 8 July 2020).

About Android App Bundles | Android Developers (2020). Available at: <https://developer.android.com/guide/app-bundle> (Accessed: 8 July 2020).

Destructuring Props in React (2018). Available at: <https://medium.com/@lcriswell/destructuring-props-in-react-b1c295005ce0> (Accessed: 8 July 2020).

Components, R., Ravi, P. and Khatri, S. (2018) *React Function Components with hooks vs Class Components*, *Stack Overflow*. Available at: <https://stackoverflow.com/questions/53062732/react-function-components-with-hooks-vs-class-components> (Accessed: 8 July 2020).

Destructuring Props in React (2018). Available at: <https://medium.com/@lcriswell/destructuring-props-in-react-b1c295005ce0> (Accessed: 8 July 2020).

Difference between TypeScript and JavaScript - GeeksforGeeks (2018). Available at: <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/> (Accessed: 8 July 2020).

Garbage collection | Cloud Bigtable Documentation | Google Cloud (2020). Available at: <https://cloud.google.com/bigtable/docs/garbage-collection> (Accessed: 9 July 2020).

Introduction to Expo - Expo Documentation (2020). Available at: <https://docs.expo.io/> (Accessed: 8 July 2020).

Nakazawa, C., McKenzie, S. and Kyle, J. (2016) *Yarn: A new package manager for JavaScript - Facebook Engineering, Facebook Engineering*. Available at: <https://engineering.fb.com/web/yarn-a-new-package-manager-for-javascript/> (Accessed: 8 July 2020).

Props · React Native (2020). Available at: <https://reactnative.dev/docs/props.html#:~:text=Most%20components%20can%20be%20customized,contro%20what%20image%20it%20shows.> (Accessed: 8 July 2020).

React: Class Component VS Function Component with Hooks - DEV (2020). Available at: <https://dev.to/danielleye/react-class-component-vs-function-component-with-hooks-13dg> (Accessed: 8 July 2020).

React-Native? W. and Wilms, J. (2019) *What is the role of Node.js in React-Native?*, *Stack Overflow*. Available at: <https://stackoverflow.com/questions/55206889/what-is-the-role-of-node-js-in-react-native#:~:text=React%20Native%20uses%20Node.,to%20build%20your%20JavaScript%20code.&text=js%20allows%20you%20to%20run,Terminal%2C%20and%20helps%20create%20modules.> (Accessed: 8 July 2020).

React Native Meets Async Functions (2016). Available at: <https://blog.expo.io/react-native-meets-async-functions-3e6f81111173> (Accessed: 8 July 2020).

Serverless Platform Comparison: Google Cloud Function vs. AWS Lambda (2019). Available at: <https://medium.com/@manus.can/serverless-platform-comparison-google-cloud-function-vs-aws-lambda-8e060bcc93b4> (Accessed: 8 July 2020).

useState and useEffect explained (2019). Available at: <https://medium.com/recraftrelic/useState-and-useEffect-explained-cdb5dc252baf> (Accessed: 8 July 2020).

useEffect? W. et al. (2018) *What's the difference between useState and useEffect?*, *Stack Overflow*. Available at: <https://stackoverflow.com/questions/53219164/what-s-the-difference-between-useState-and-useeffect#:~:text=UPDATE-,what's%20the%20exact%20difference%20between%20useState%20and%20useEffect%20%3F,only%20supported%20for%20class%20components.> (Accessed: 8 July 2020).

var vs let vs const in JavaScript - UI.dev (2020). Available at: <https://ui.dev/var-let-const/#:~:text=Turns%20out%2C%20const%20is%20almost,it%20to%20a%20new%20value.&text=The%20take%20away%20above%20is,with%20const%20can't%20be.> (Accessed: 8 July 2020).

What are the main differences between ReactJS and React-Native? (2019). Available at: <https://medium.com/@alexmngn/from-reactjs-to-react-native-what-are-the-main-differences-between-both-d6e8e88ebf24> (Accessed: 8 July 2020).

Which is Better React JS Or React Native for Mobile Apps Development? - GoodWorkLabs (2017). Available at: <https://www.goodworklabs.com/react-native-versus-react-js/> (Accessed: 8 July 2020).

webpart? A. and Randal, J. (2019) *Anyone can explain the .ts file and .tsx file use and difference in SPFx react webpart?* *SharePoint Stack Exchange*. Available at: <https://sharepoint.stackexchange.com/questions/263420/anyone-can-explain-the-ts-file-and-tsx-file-use-and-difference-in-spfx-react-w#:~:text=TSX%20allows%20for%20embedding%20JSX,presentation%20in%20the%20TSX%20files.> (Accessed: 8 July 2020).

YARN vs. NPM (2019). Available at: <https://medium.com/@j.dumadag718/yarn-vs-npm-b2d58289fb9b> (Accessed: 8 July 2020).