ELECTRON FRAMEWORKS

ABOUT ELECTRON:

- Electron is a free and open source framework which is used to create desktop applications using web technologies.
- Electron is based on the open source web browser project Chromium (CHROMIUM)
- In Electron, the main process is in charge of producing the main window and handling the application life cycle.
- A renderer process is in charge of rendering the UI of an Electron application.
- Electron uses IPC (Inter process communication) to communicate between the main process and the renderer processes.

METHODS:

- server.listen method begins listening for incoming connections on the port and host supplied.
- Net.createServer method builds a new network server that can accept connections.
- To quit the application App.quit() method is used.

FUNCTIONS:

- Various functions used in electron:-

i) fs.readFile(filePath, options, callback) — reads file's contents.

ii) IpcRenderer.send — delivers an IPC message from renderer process to main process.

iii) event.sender.send — transmits an IPC message from one renderer process to another.

iv) desktopCapturer.getUserMedia — returns audio and video streams for chosen source.

→ Difference between React Native & React:

i) React is used for building web applications, while React Native is used for building native mobile apps.

ii) React uses virtual DOM to update the UI, while React Native uses native components instead of virtual DOM/DOM.

iii) React uses CSS for styling, while React Native uses styling system similar to CSS but not identical.

iv) React has React-router for navigating web pages while React Native has built-in Navigator for navigating mobile apps.

→ STYLING AND LAYOUT IN REACT NATIVE:

Stylesheets in React Native are written in Javascript, rather than CSS. React Native includes set of layout & positioning tools, such as flexbox, which help in arranging and size the components.

→ FLEXBOX:

Flexbox, short for Flexible Box Layout, is a layout model which provides flexible & scalable way of arranging UI elements in a 2-D space. It enables user to specify the direction of the main axis (row/column), then align and distribute space among elements along main axis.

→ Debugging Tools for React Native:

i) Javascript Debugging
ii) Logging
iii) React Developer Tools.
iv) Hot Reloading.
v) Remote Debugging

React Native Bridge:

React Native Bridge is a mechanism for communication between Javascript code & native platform code. Javascript code runs on Javascript Thread whereas native code runs on main thread, hence react native bridge responsible for passing data and messages among above two.

Bridge allows Javascript code to call native platform APIs and hence access native functionality, such as camera, accelerometer & other device APIs as well to display native components such as buttons, text inputs, etc.

Threads run in React Native :-

React Native UI thread (main thread) – for layout of mobile App.

React Native Javascript Thread. – for the business logic execution

React Native Modules Thread – To access platform API.

React Native Render Thread – To generate commands which is used to draw app UI.

Steps to create and start React Native App :-

step ① - Install node.js.

step ② - Install react-native environment by command.

```
$ npm install -g create-react-native-app
```

step ③ - create project using command.

```
$ create-react-native-app myProject.
```

step ④ - navigate to your project.

```
$ cd myProject.
```

step ⑤ - Now run following command to start project.

```
$ npm start.
```