**Section 1 Summary:**

Web XR offers an opportunity to place 3D models into your home whilst shopping online e.g on Shopify. You can test how a vase looks on your table and then purchase it if it's your cup of coffee.

It’s an interesting technology that is based off of threejs.ThreeJS is a javascript library which is used to build models and enhance real world experiences. It's also fun when implemented in a great environment.

It requires some creativity and curiosity about how it enhances life. It’s quite fun actually😂. So, obviously WebXr developers have a lot of fun.

To be proficient you need to have great JS skills. You also need to be proactive. This is cutting edge technology. It is immensely used on platforms such as IOS , Android and on Desktop.

Let’s move on to section 2.

**Section 2 Summary:**

**WebXR**

* Recognises compatible XR/VR devices.
* Manages the timing, scheduling, and the various points of view relevant when drawing a 3D scene, especially the correct frame rate.
* Access to all sensors on your device. Hence the ability to sense the movement of the device.
* Support for accessing inputs from control devices( like handheld VR controllers or mixed reality gamepads).

Sensors everywhere

Inertial Measurement Unit

6 degrees of freedom

WebXR does not load or manage or render 3D models

WebGL and Three.js are the ones that do the 3D models

**WebXR Application lifecycle**

**Controller**

**mesh.quartenion**

**Detect that there has been a tap**

**Get info about position and rotation of that tap using matrixWorld.**

**Already drawn object should be in relation to a space that is not changing so that they maintain position.**

**We are using these spaces to get coordinates of different models that we will draw.**

**ViwerSpace - moves with the phone.**

**localSpace is used to get Coordinates of different things that we have drawn. Starts when the ar session begins.**

**Hit**

**Pose-position and orientation of the point on the space**

**Translate the position and orientation information from our space from the pose onto the Reticle**

Matrix is an object that contains info about the position and orientation of something in 3D

We are creating an hit test and we are finding a point on the surface which has a pose

A pose being an object that contains the position and orientation of that point and

We are translating the information on that pose onto the reticle and placing the reticle onto the same space;