LAB 8: Create a Marker less AR to display 3D model in website or web-based application

Sol:

Step 1: download .Patt file and same .jpg file

https://github.com/jeromeetienne/AR.js/blob/master/three.js/examples/marker-training/examples/pattern-files/pattern-letterA.patt

Step 2: Write JavaScript and HTML code index.HTML

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Include A-Frame -->
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>

    <!-- Include AR.js for A-Frame -->
    <script src="https://cdn.jsdelivr.net/gh/jeromeetienne/ar.js/aframe/build/aframe-
ar.min.js"></script>
  </head>
  <body style="margin: 0px; overflow: hidden;">
    <a-scene embedded arjs>
      <!-- Marker -->
      <a-marker type="pattern" url="pattern-letterA.patt">
        <a-box position="0 0.5 0" material="opacity: 0.5;"></a-box>
        <a-cylinder color="green" height="1.0" radius="0.5" position="1 0.5 0"></a-cylinder>
      </a-marker>

      <!-- Camera -->
      <a-entity camera></a-entity>
    </a-scene>
  </body>
</html>
```

Step 3: create server or deploy application for testing

To create Express server

1. Create Project

```
2. mkdir my-aframe-project
3. cd my-aframe-project
4. npm init -y
5. create folder public inside project folder
```

2. Install express
   To check if already install:
   npm list express

```
npm install express
```

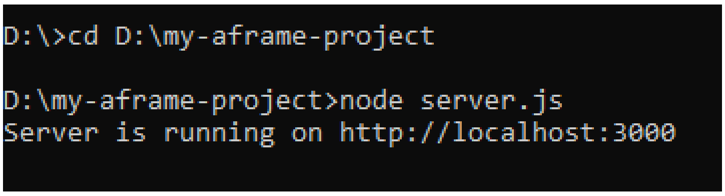
3. Create server.js script for creating server

```
const express = require('express');
const path = require('path');
const app = express();

// Serve static files from the "public" directory
app.use(express.static(path.join(__dirname, 'public')));

// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

4. Start Your Server

```
node server.js
```

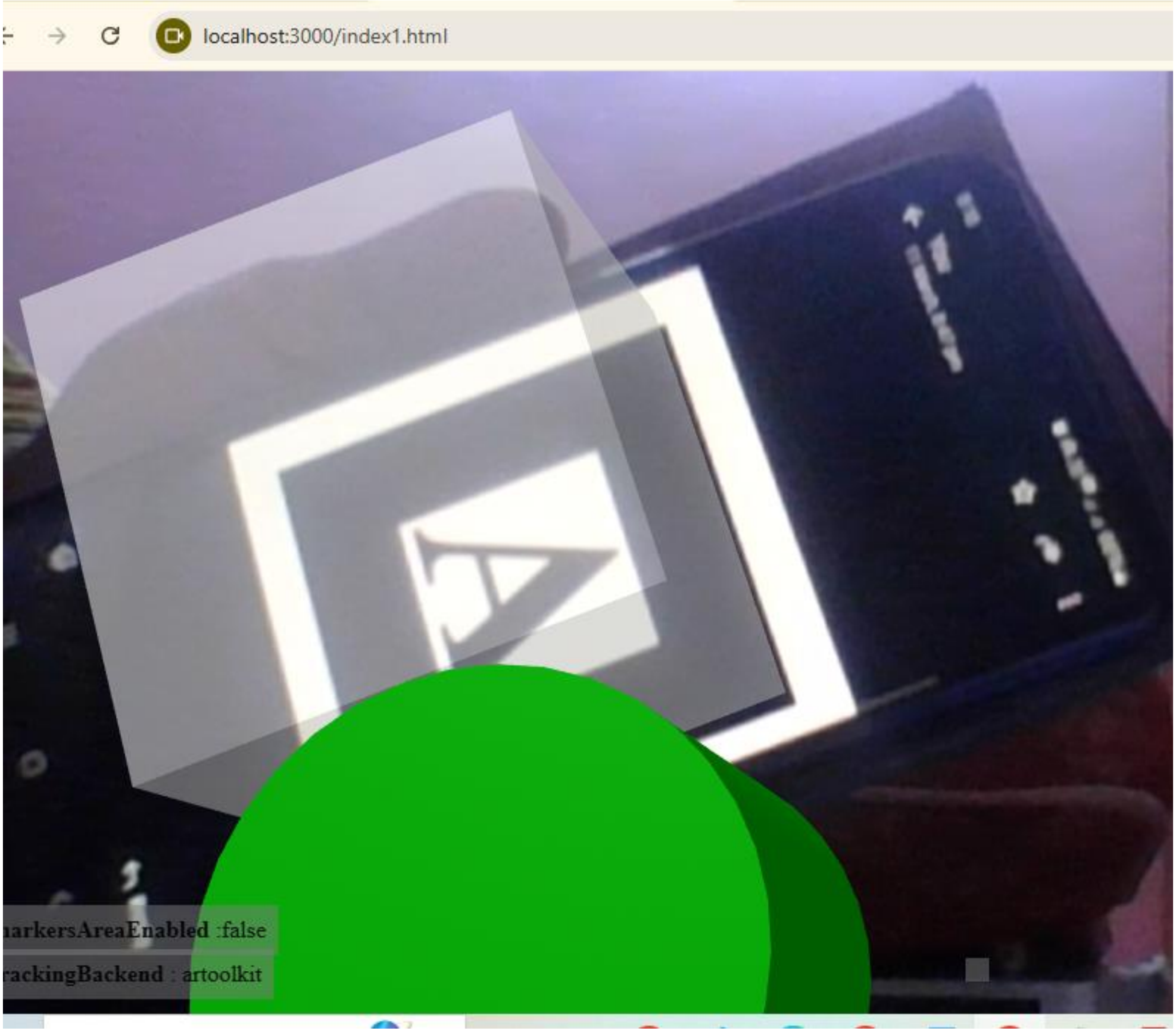
//here you need to use command prompt ->locate your folder and run server



```
D:\>cd D:\my-aframe-project

D:\my-aframe-project>node server.js
Server is running on http://localhost:3000
```

5. Type http://localhost:3000/index.html in the browser to run application

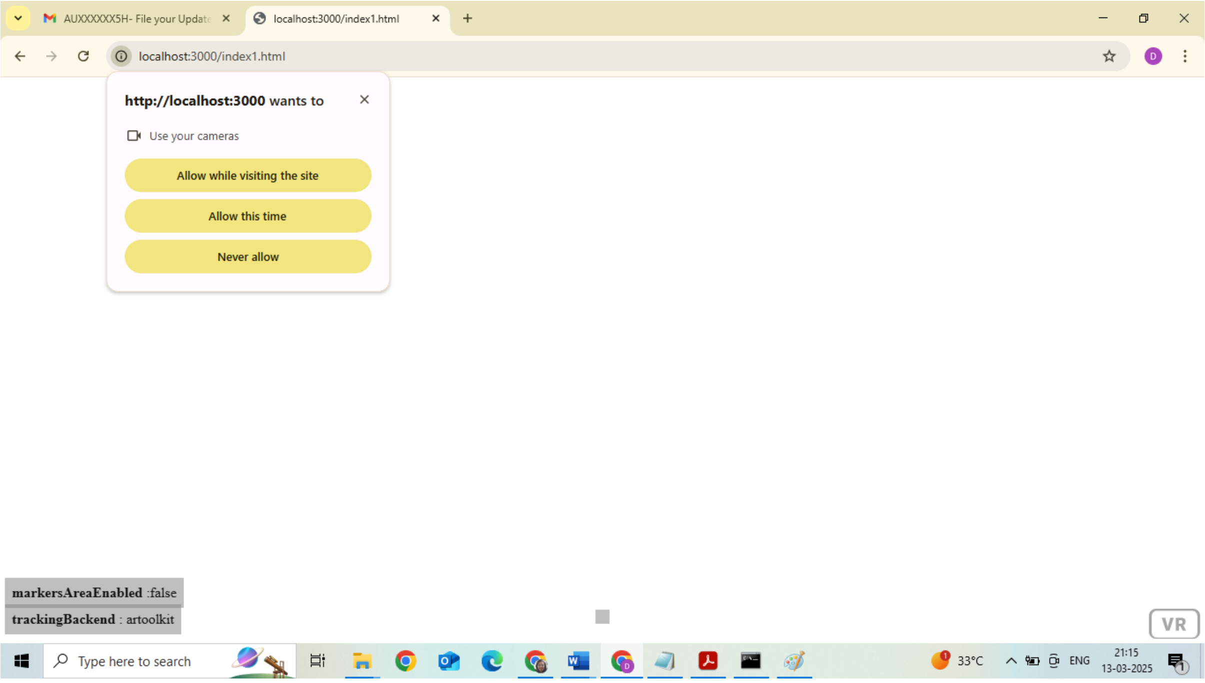
o/p:



localhost:3000/index1.html

arkersAreaEnabled :false

rackingBackend : artoolkit



AUXXXXXX5H- File your Updat
localhost:3000/index1.html
+
localhost:3000/index1.html
D

http://localhost:3000 wants to

Use your cameras

Allow while visiting the site
Allow this time
Never allow

markersAreaEnabled :false

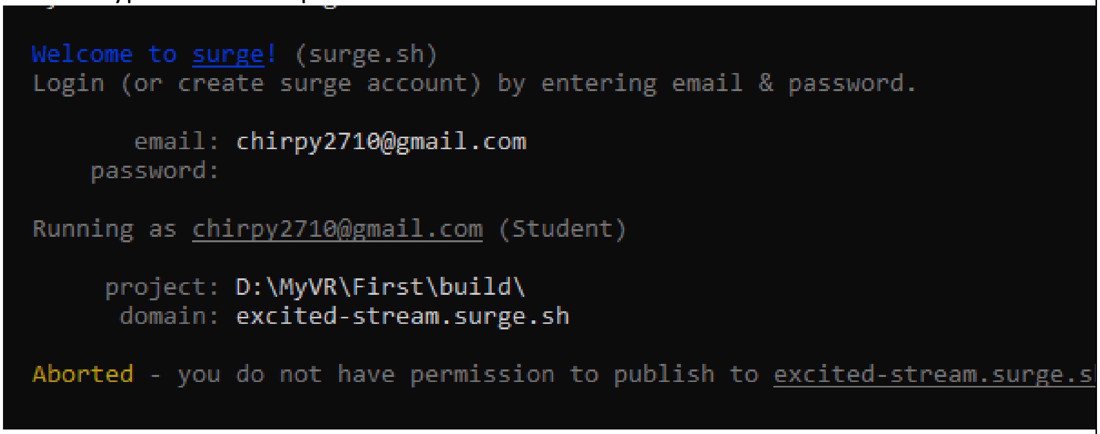
trackingBackend : artoolkit

VR
Type here to search
33°C
ENG
21:15
13-03-2025

Or Surge deployment if no web cam

//Deployment using surge

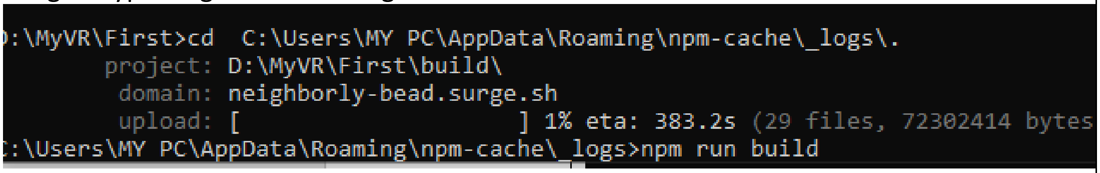
1. >npm install -g surge
2. >surge –version
3. >npm run bundle
4. In main folder check build file
5. Copy static_asset in build folder
6. Cd build
7. Type surge
8. Type enter
9. Type mail id and password



```
   Welcome to surge! (surge.sh)
   Login (or create surge account) by entering email & password.

          email: chirpy2710@gmail.com
       password:

   Running as chirpy2710@gmail.com (Student)

        project: D:\MyVR\First\build\
         domain: excited-stream.surge.sh

   Aborted - you do not have permission to publish to excited-stream.surge.s
```

10.Verify account

11. gain type surge then enter again enter



```
:\MyVR\First>cd  C:\Users\MY PC\AppData\Roaming\npm-cache\_logs\.
       project: D:\MyVR\First\build\
        domain: neighborly-bead.surge.sh
        upload: [                    ] 1% eta: 383.2s (29 files, 72302414 bytes
:\Users\MY PC\AppData\Roaming\npm-cache\_logs>npm run build
```

Domain is : neighborly-bead.surge.sh

For latest node version

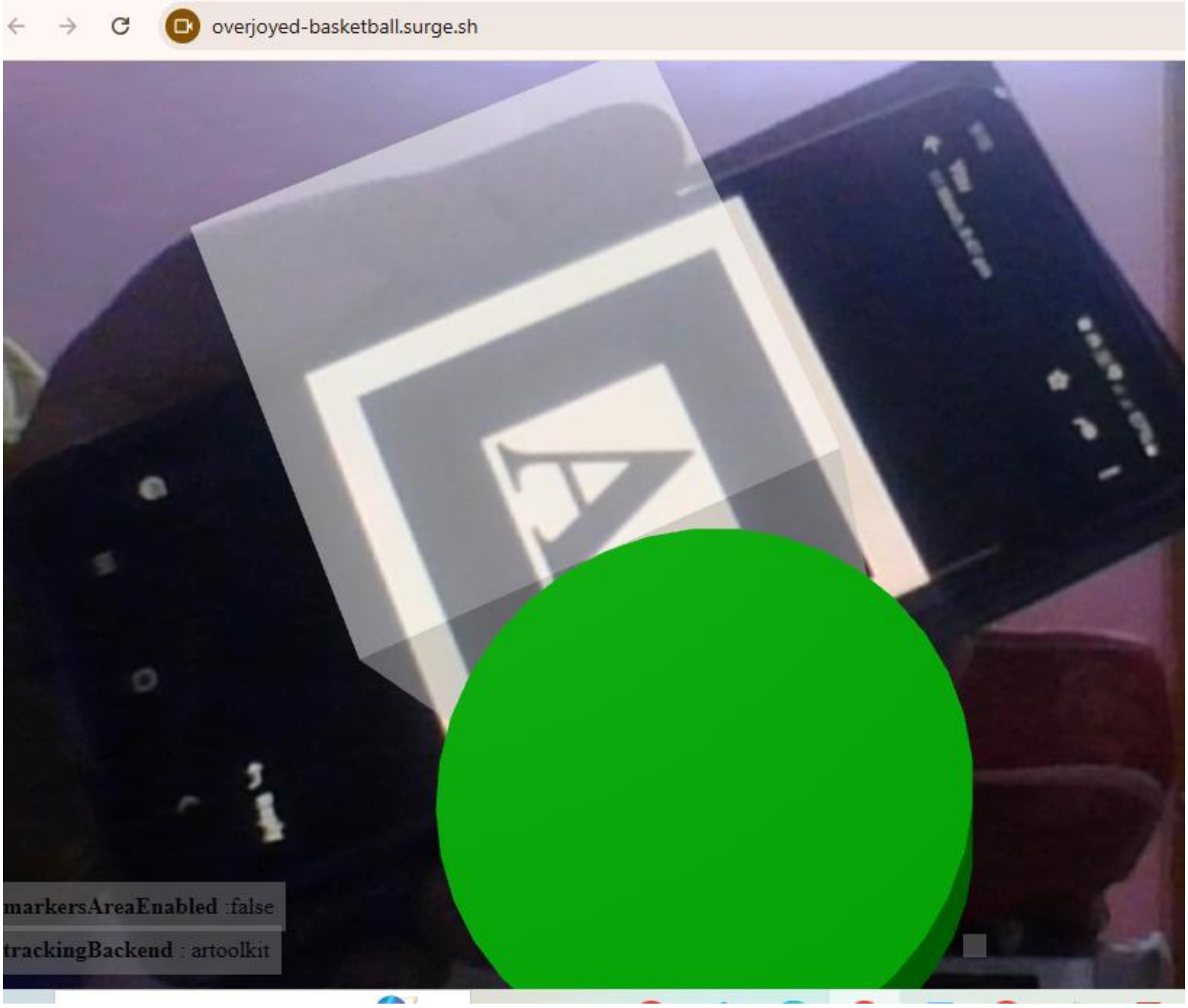
- npm init -y
- npm update
- npm install surge@latest --save
- webpack file
- index.js
- npm install webpack-cli --save-dev
- npm install html-webpack-plugin --save-dev
- npm run bundle or npx webpack
- Type surge

Edit: package.json

Go to build folder dist from cmd

Type surge and copy .sh URL with https://

Output



markersAreaEnabled :false

trackingBackend : artoolkit