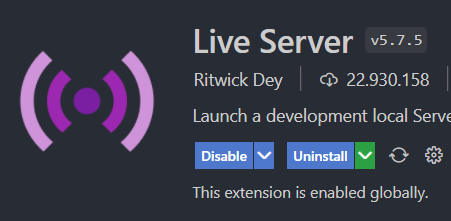
**HƯỚNG DẪN BUILD VÀ THỰC THI PROJECT**

I. Chuẩn bị các ứng dụng và tài nguyên cần thiết

\* Cài đặt Visual Studio Code



\* Cài đặt môi trường Live Sever cho Visual Studio Code để chạy trang web



\* Tải tài nguyên hỗ trợ tạo các model

+ three.js: truy cập đường link <https://github.com/mrdoob/three.js/blob/dev/build/three.js>

Ảnh có chứa văn bản

Mô tả được tạo tự động

II. Tạo File và thiết lập trang web

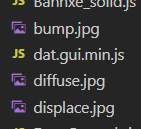
**\*Thêm các file để hỗ trợ:**

- image: trang trí tạo biểu tượng cho các Model

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Các file này nên để ngoài không nên để folder vì trong lúc code nhóm em để link truy cập đến file không có folder.



* Các file hình này dowload ở link sau:

<https://drive.google.com/drive/folders/1LJRK7R8SqyN9Pdbp4eQ6hWnLOVSTKZKM>

- Các file js: dùng để vẽ theo loại Model, theo kiểu vẽ,(File OrbitControls1 là file OrbirControls hỗ trợ module trong three.js-dev đổi tên để tránh trùng).

* Các file Shepre\_edge.js, Shepre\_point.js, Shepre\_solid.js, tương tự các file có hậu tố edge, point, solid đều là tự code.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Ngoài những file này ra còn có









Đây là những file js cần để thực hiện project.

**\* Code file js có hậu tố edge, point, solid.**

**Theo yêu cầu của đồ án nhóm em làm 22 file .js để vẽ các khối hình:**

**Gồm: Amtra\_.js: 3 file vẽ ấm trà theo edge, point, solid**

**Banhxe\_.js: 3 file vẽ bánh xe theo edge, point, solid**

**Sphere\_.js: 3 file vẽ hình cầu theo edge, point, solid**

**TorusKnot\_.js: 3 file vẽ hình tự tìm hiểu theo edge, point, solid**

**Hinhtru\_.js: 3 file vẽ hình trụ theo edge, point, solid**

**Hinhnon\_.js: 3 file vẽ hình nón theo edge, point, solid**

**Hinhhop\_.js: 3 file vẽ hình hộp theo edge, point, solid**

**Loadmodel.js: 1 file vẽ model tải về trên mạng (chỉ có thể vẽ được model ra theo kiểu solid)**

**Ngoại trừ các file Amtra\_.js và file Loadmodel.js thì các file còn lại code theo kiểu như sau:**

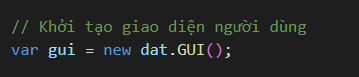
* **Khởi tạo hàm init():**



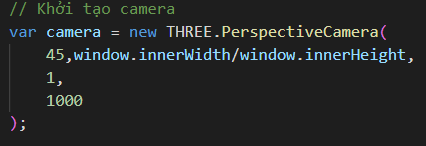
* **Các bước sau sẽ được viết trong hàm init():**
  + **Khởi tạo cảnh scene:**



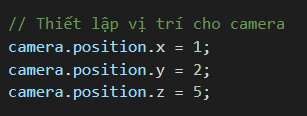
* + **Khởi tạo giao diện người dùng**



* + **Khởi tạo camera:**



* + **Thiết lập vị trí cho camera:**



* + **Thiết lập điểm nhìn của camera**



* + **Khởi tạo khối hình (ví dụ hình nón vẽ theo edge):**



* + - **Các thông số được truyền vào là các thông số khởi tạo hình dạng của khối hình (truy cập vào đường link** [**https://threejs.org/docs/#api/en/geometries/ConeGeometry**](https://threejs.org/docs/#api/en/geometries/ConeGeometry)

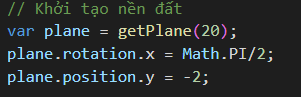
**Để biết thêm chi tiết, có thông tin các khối hình trong link)**

* + - **Vẽ theo point, solid thì gọi hàm khác nhau ở bước này:**

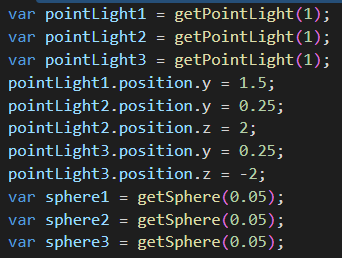




* + - **Hàm vẽ khối hình sẽ được viết sau:**
  + **Khởi tạo nền đất:**



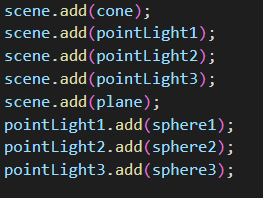
* + **Khởi tạo và đặt vị trí cho ánh sáng:**



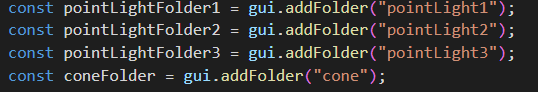
* + **Đặt tên cho biến lưu khối hình:**



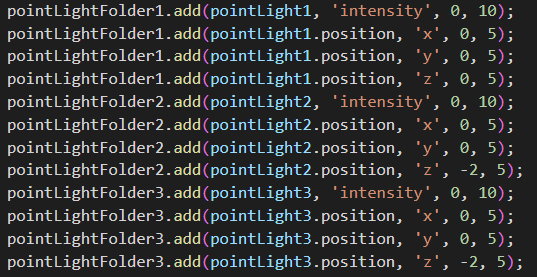
* + **Thêm các đối tượng vào cảnh:**

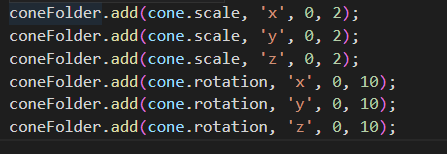


* + **Tạo các folder trong giao diện người dùng:**



* + **Thêm các thuộc tính vào các folder giao diện:**





* + **Khởi tạo trình kết xuất:**



* + **Kích hoạt thuộc tính kết xuất bóng của trình kết xuất**



* + **Thiết lập kích thước của trính kết xuất:**



* + **Đẩy thuộc tính domElement của renderer vào thẻ webgl trong file html:**



* + **Khởi tạo biến điều khiển camera:**



* + **Khởi tạo biến điều khiển object:**



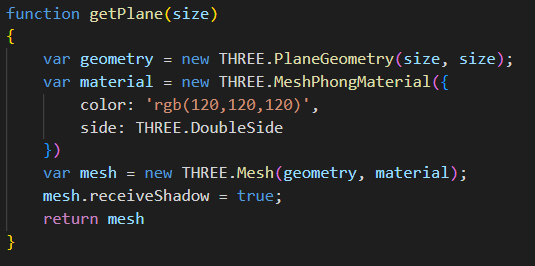
* + **Gọi hàm update để kết xuất hình ảnh liên tục:**



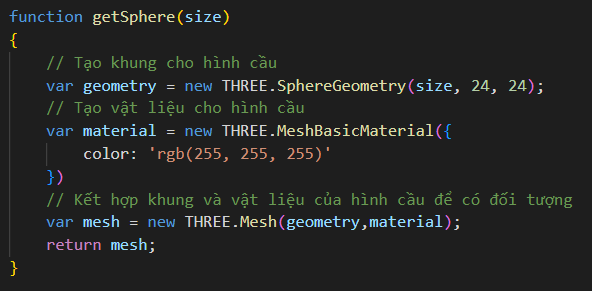
* + **Cuối hàm trả về biến scene**



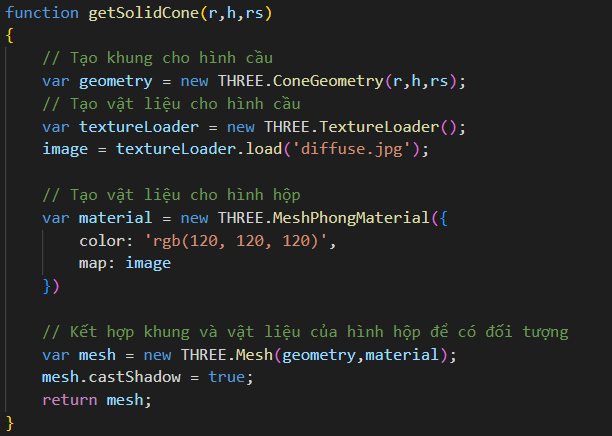
* + **Các hàm viết hỗ trợ hàm init():**
    - **Hàm getPlane(size):**



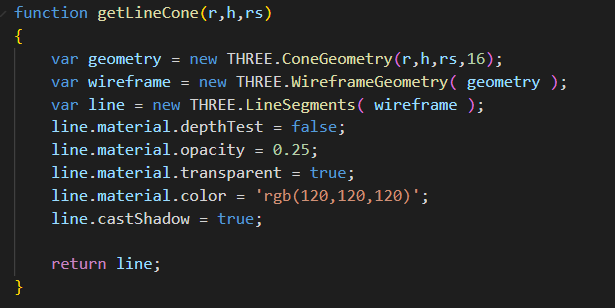
* + - **Hàm getSphere(size):**



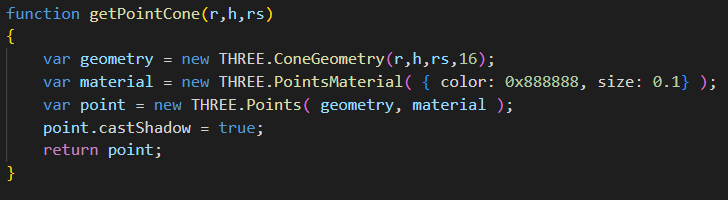
* + - **Các hàm vẽ khối hình( mỗi hàm sẽ được sử dụng ở file .js tương ứng)**
      * **Solid:**



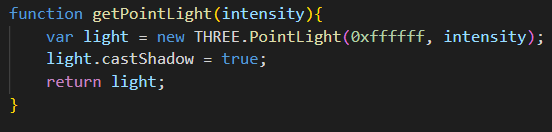
* + - * **Edge:**

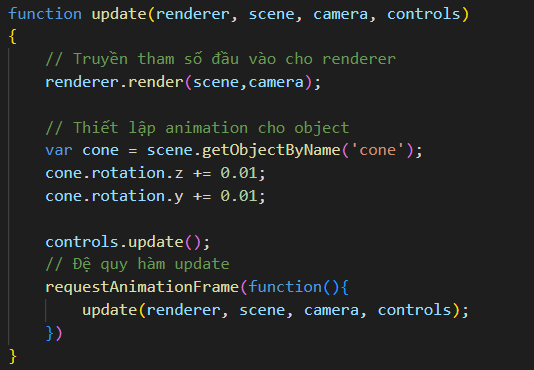


* + - * **Point:**



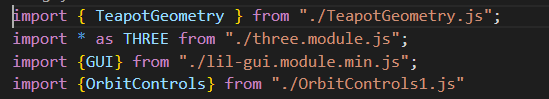
* + - **Hàm getPointLight(intensity):**



* + - **Hàm update(renderer, scene, camera, controls):**
  + **Sau khi viết hết các hàm thì khởi tạo biến scene khởi động hàm init() :**
    - **Viết bên ngoài và sau tất cả các hàm:**



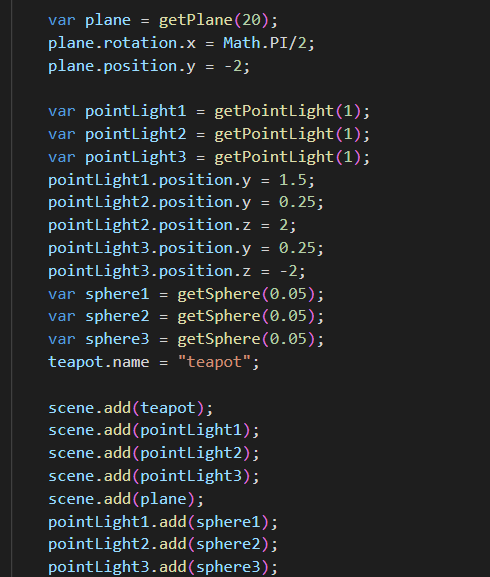
* **Các file Amtra\_.js và loadmodel.js bọn em viết theo file loại module nên sẽ có một số sửa đổi:**
  + **Các file Amtra\_.js:**
    - **Trước tiên sẽ import các tài nguyên cần thiết vào:**



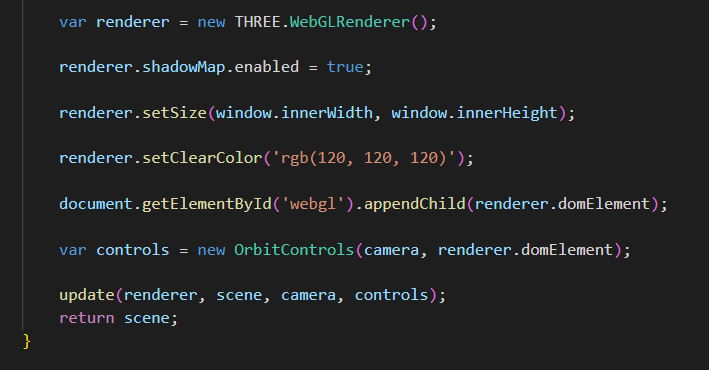
**Các file này tất nhiên đã được gắn vào folder project**

* + - **Sau đó tương tự như các file khác chỉ là có một số thay đổi:**
      * **Hàm init():**

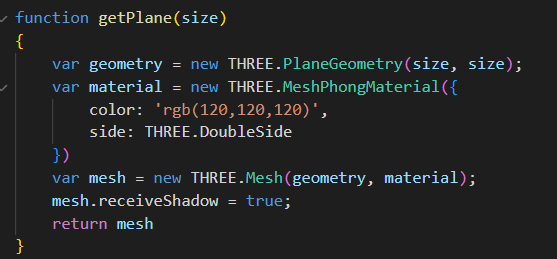




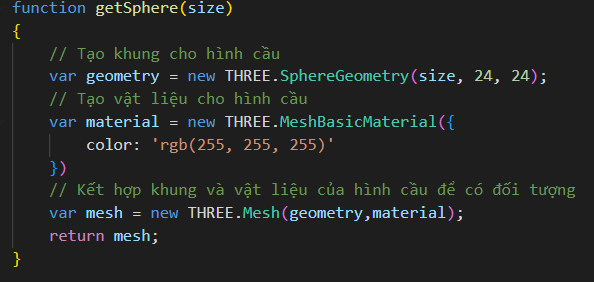




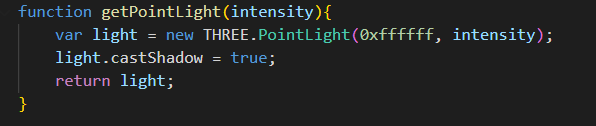
* + - * **Hàm getPlane(size):**



* + - * **Hàm getShpere(size):**



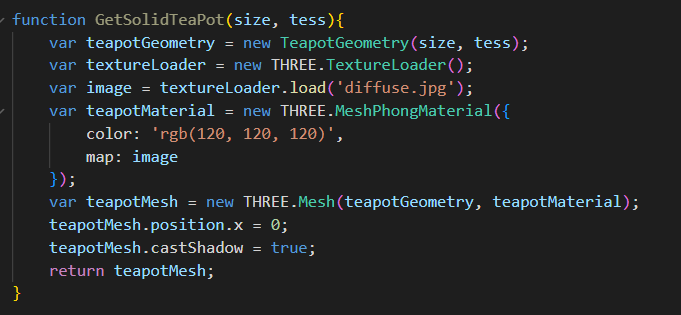
* + - * **Hàm getPointLight(intensity):**



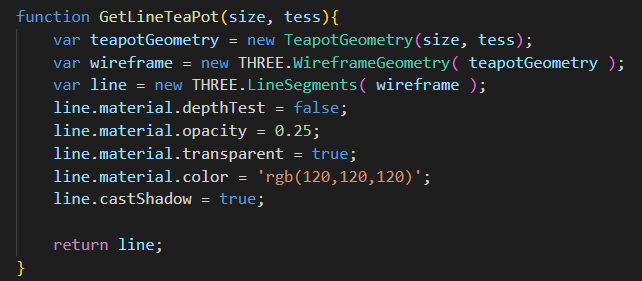
* + - * **Hàm update(renderer, scene, camera, controls):**



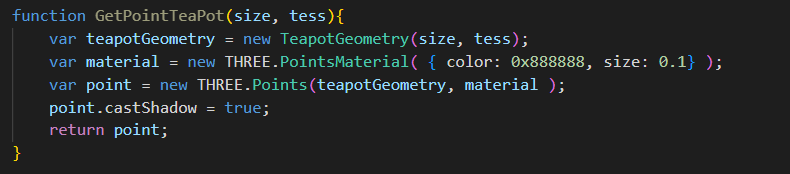
* + - * **Các hàm solid, point, line vẫn sẽ được dùng ở các file tương ứng:**
        + **Solid**



* + - * + **Line:**



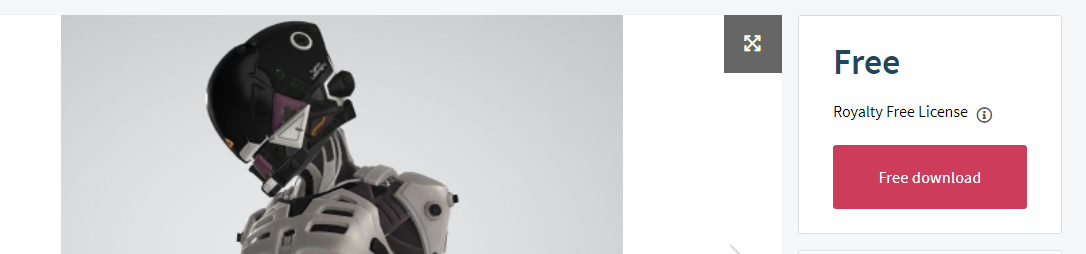
* + - * + **Point:**



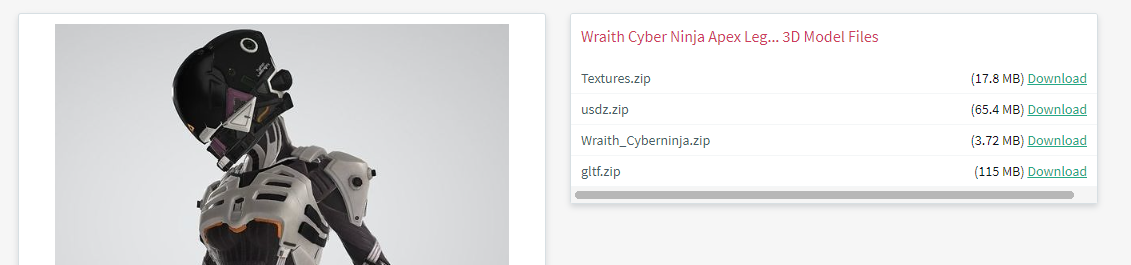
* + **Khởi động hàm init và gắn biến:**



* **File loadmode.js:**
  + **Truy cập vào link: https:**
    - **//www.cgtrader.com/free-3d-models/character/woman/wraith-cyber-ninja-apex-legends**
    - **Tải mô hình về:**



* + - **Sau khi bấm dowload sẽ chờ vài giây rồi sẽ hiện lên như sau:**

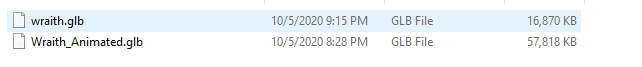


* + - **Tải file gltf.zip về:**
    - **Sau khi tải về giải nén ra và truy cập vào file gltf:**

**Em để nó ở file dowload luôn nên sẽ có đường dẫn như sau:**



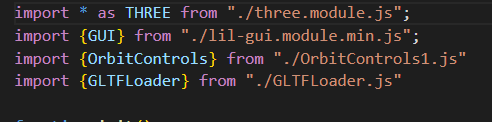
**Có 2 file trong thư mục:**



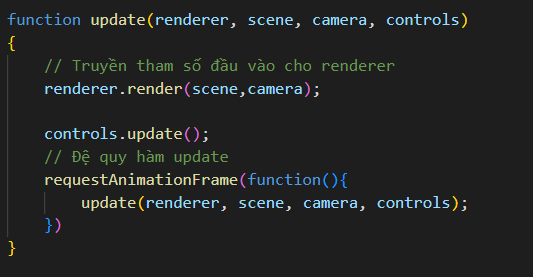
**Tạo một folder mới trong file project đặt tên là “assets”:**

**Kéo file wraith.glb vào sau đó thực hiện các bước tiếp theo**

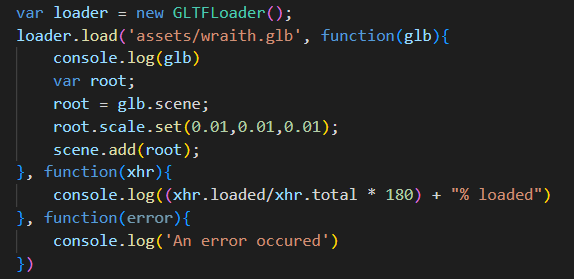
* + **Import các tài nguyên cần thiết:**



* + **Không có các hàm \_solid, \_line, \_point:**
  + **Hàm getPlane, getSphere, getPointLight: giống với các file Amtra\_.js:**
  + **Hàm update bỏ bớt 1 vài thứ:**

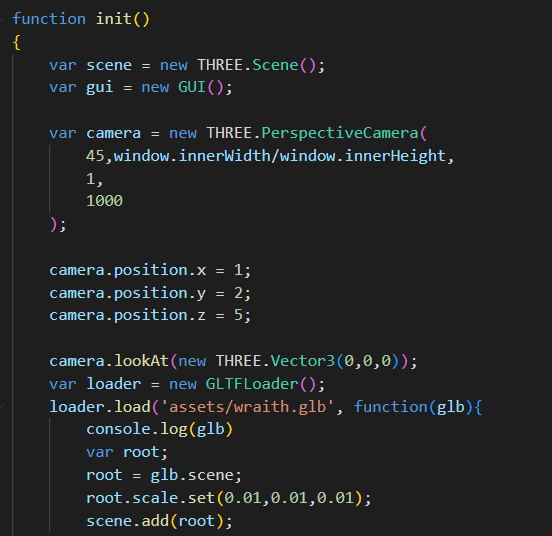


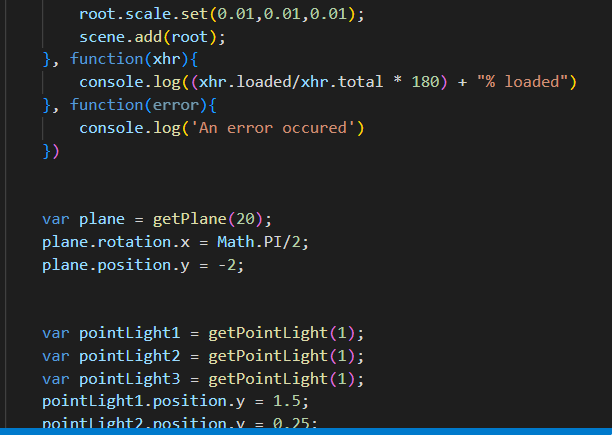
* + **Hàm init bỏ đi bước khởi tạo khối hình thay bằng load model từ file tải về:**



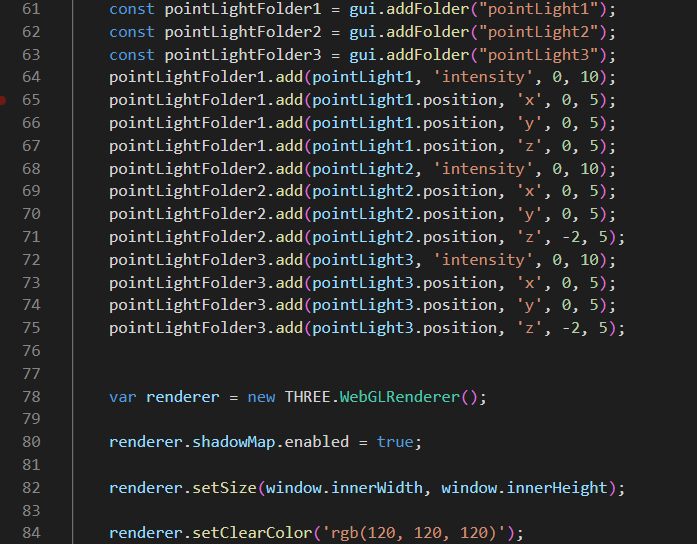
**Bỏ đi biến điều khiển đối tượng:**

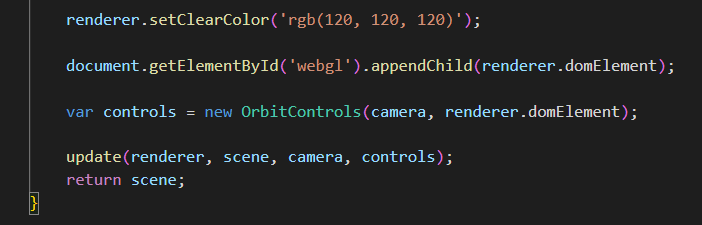
* + **Hàm init() trọn vẹn của loadmodel.js:**











**\* Tạo file trang chủ dùng để tổng hợp các model**



**\* Tiến hành thiết lập các chức năng cho web trang chủ:**

- Chọn loại hình để vẽ bao gồm **list**: Hình hộp, Hình cầu, Bánh xe,vvv

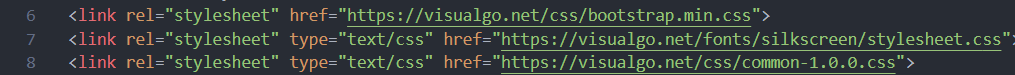
Vd: Đối với **Hình hộp**

Ảnh có chứa văn bản, màn hình, ảnh chụp màn hình

Mô tả được tạo tự động

* Như Model Hình hộp ta tiến hành tương tự với các Model khác.

+Thêm các file css có sẵn

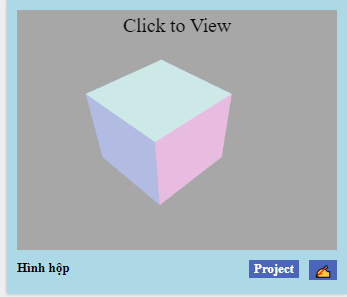


+Trang trí thêm cho các biểu tượng trong trang web:

Ảnh có chứa văn bản

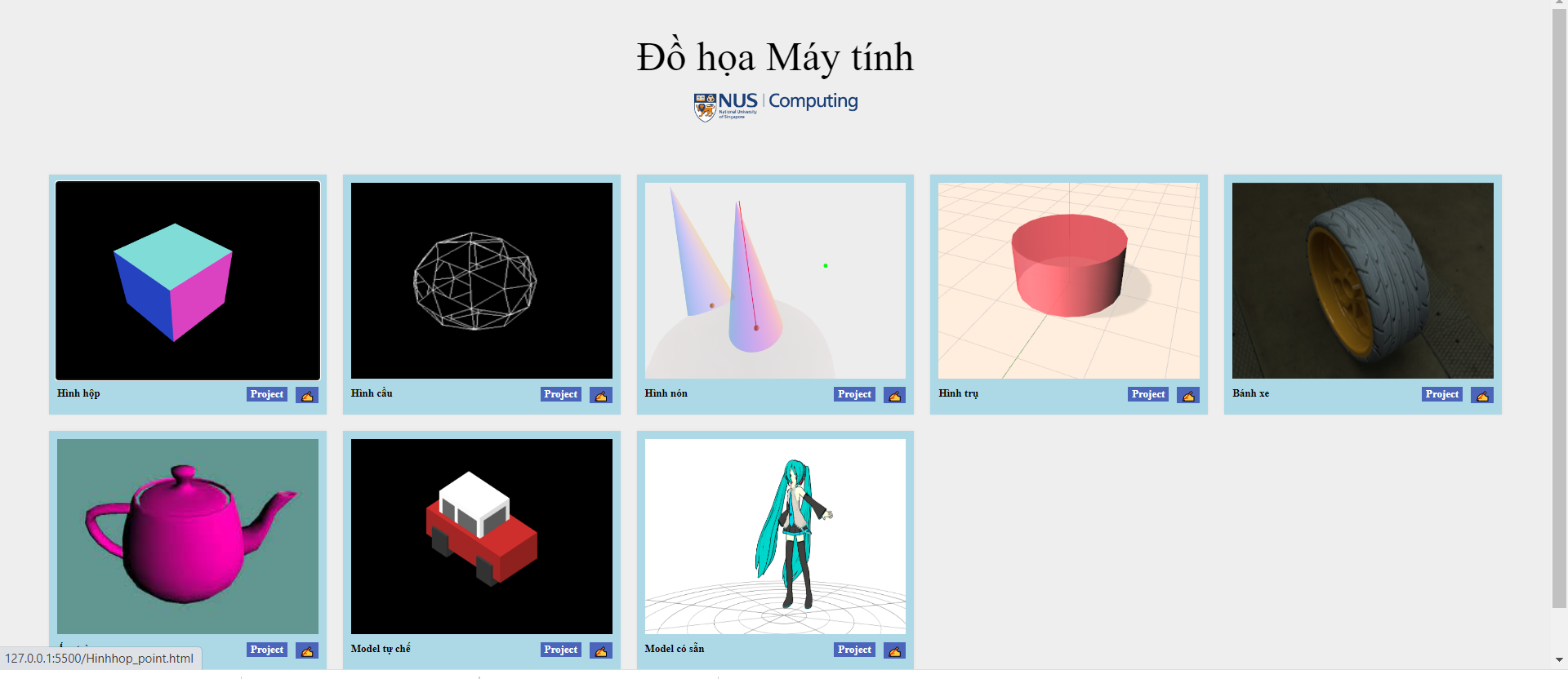
Mô tả được tạo tự động

* Kết quả

*Biểu tượng khi chưa trỏ chuột*

*Biểu tượng khi trỏ chuột*



*Giao diện trang web* ***index.html***

**\* Tạo các trang web chứa các Model được vẽ theo các kiểu**

Ảnh có chứa văn bản

Mô tả được tạo tự động

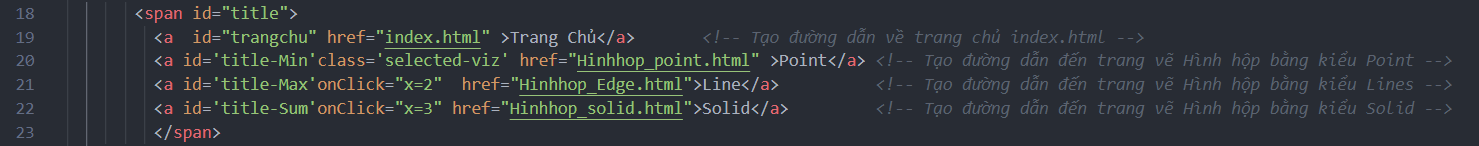
+ File có: đuôi ***\_edge*** vẽ hình theo kiểu **Lines**

: đuôi ***\_point*** vẽ hình theo kiểu **Point**

: đuôi ***\_solid*** vẽ hình theo kiểu **Solid**

**\* Thiết lập các chức năng cho các trang web Model:**

Vd: đối với file **Hinhhop\_point.html**



\* Thêm các file js để vẽ các Model cũng như hỗ trợ xem:

Ảnh có chứa văn bản

Mô tả được tạo tự động

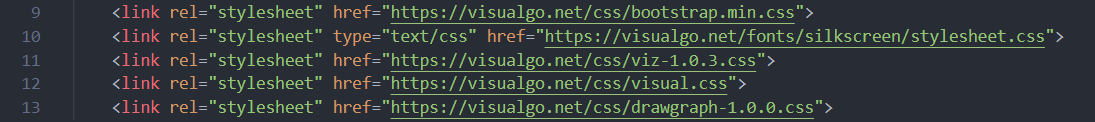
\* Đối với từng web Model ta thiết lập một file js tương ứng

vd: +Trang web **Hinhhop\_point.html** sử dụng js:

+Tương tự web **Amtra\_solid.html** sử dụng js:



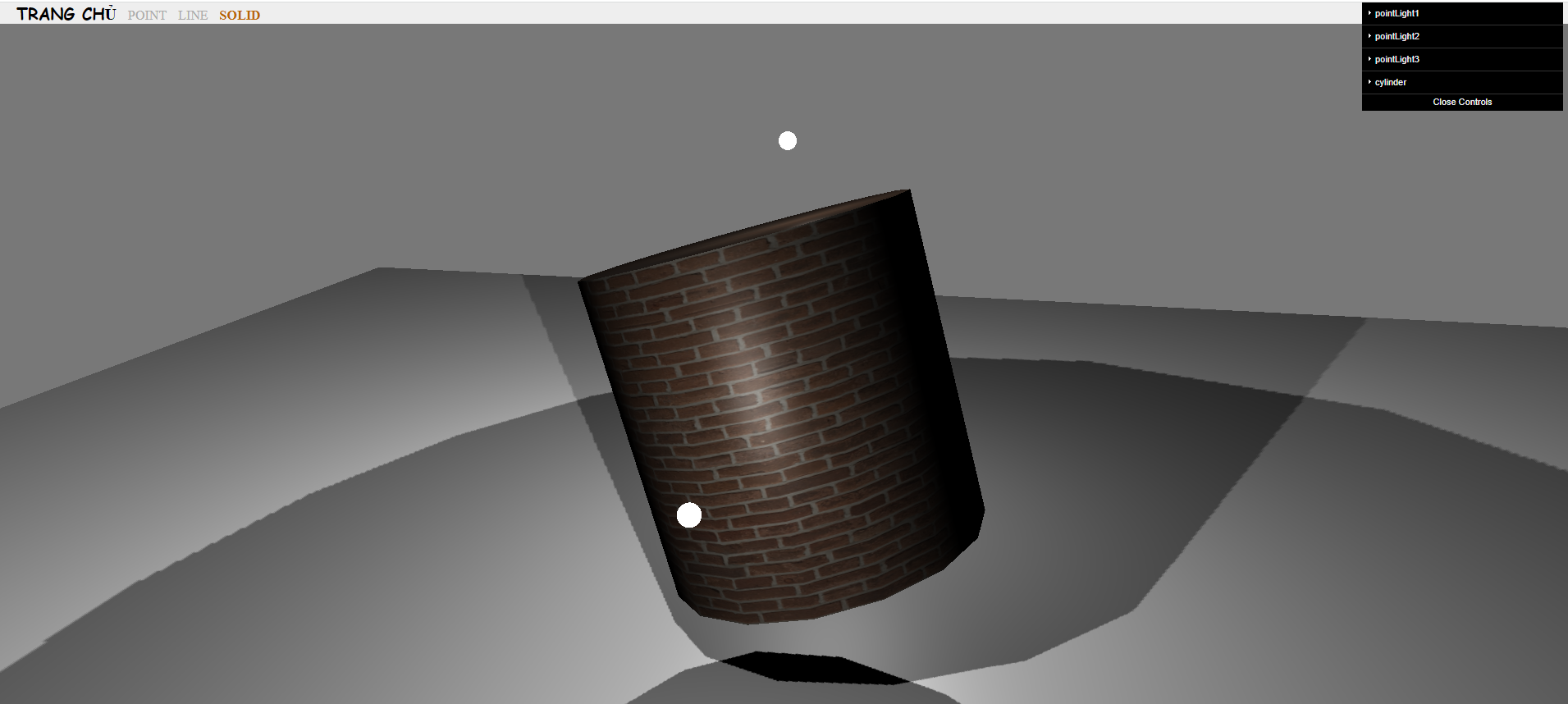
\*Trang trí các trang web Model bằng các file css có sẵn và main.css





→Kết quả:

Vd: đối với web Hinhtru\_solid.html



-Các model khác build tương tự

III. Thực thi trang web

-Mở Visual Studio Code ->File -> Open Folder-> “Chọn file chứa đồ án”

-Mở file **index.html**

Ảnh có chứa văn bản

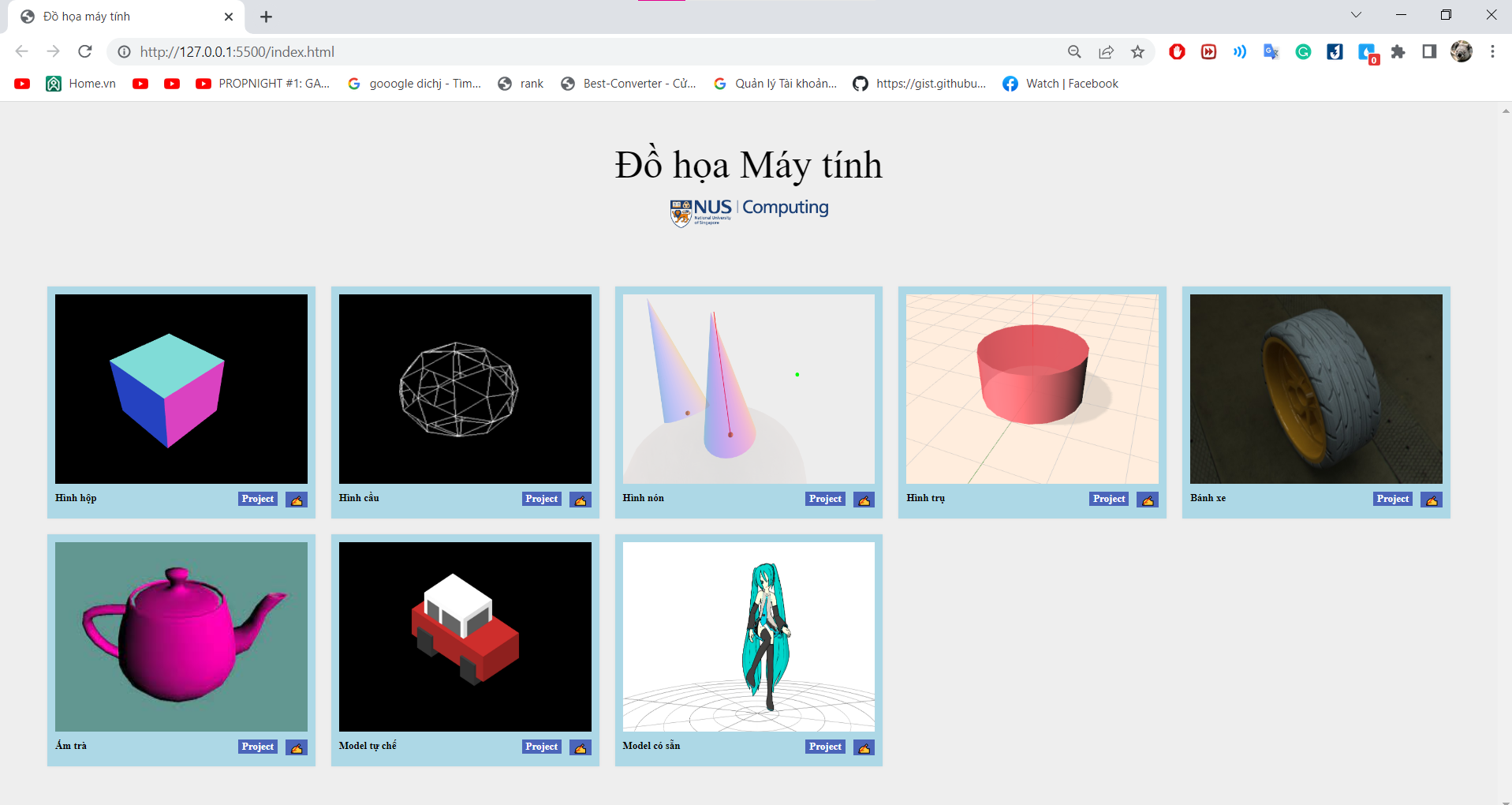
Mô tả được tạo tự động

-Sau đó nhấn chuột phải chọn Open with Live Sever để chạy trang web

Ảnh có chứa văn bản

Mô tả được tạo tự động

- Kết quả:

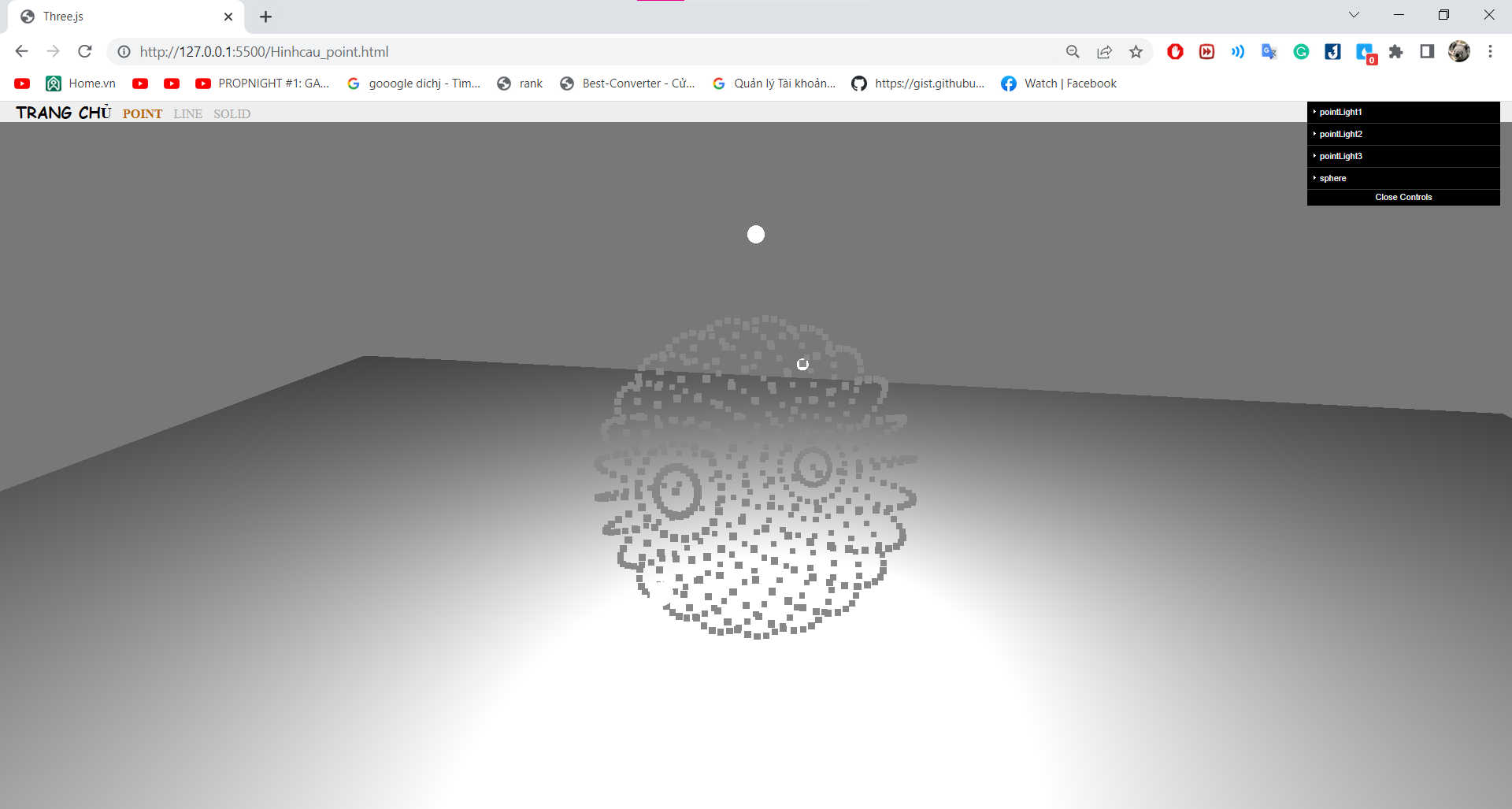


- Chọn biểu tượng Model bất kỳ giao diện sẽ chuyển sang trang web chứa Model và mặc định Model vẽ theo kiểu Point

+ví dụ như Chọn Hình cầu

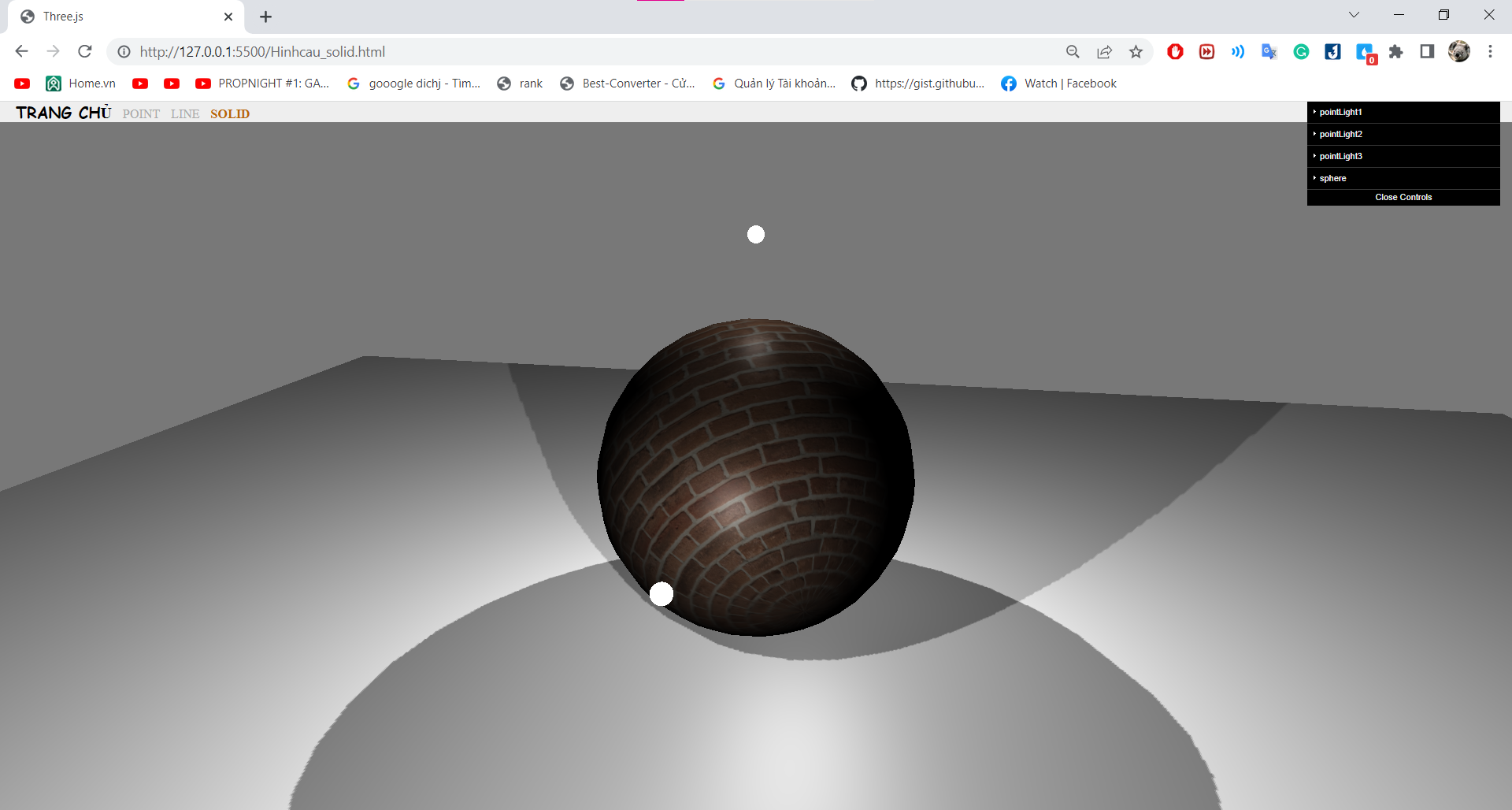
Ảnh có chứa văn bản, các môn điền kinh, thể thao, bóng rổ

Mô tả được tạo tự động

 - Nếu muốn xem Model vẽ theo kiểu khác có thể nhấn vào Lines hoặc Solid

Ảnh có chứa văn bản, trong nhà, ảnh chụp màn hình

Mô tả được tạo tự động



- Muốn chọn lại các Model khác có thể nhấn nút trang chủ để về lại trang web ban đầu

