What we aim to achieve:

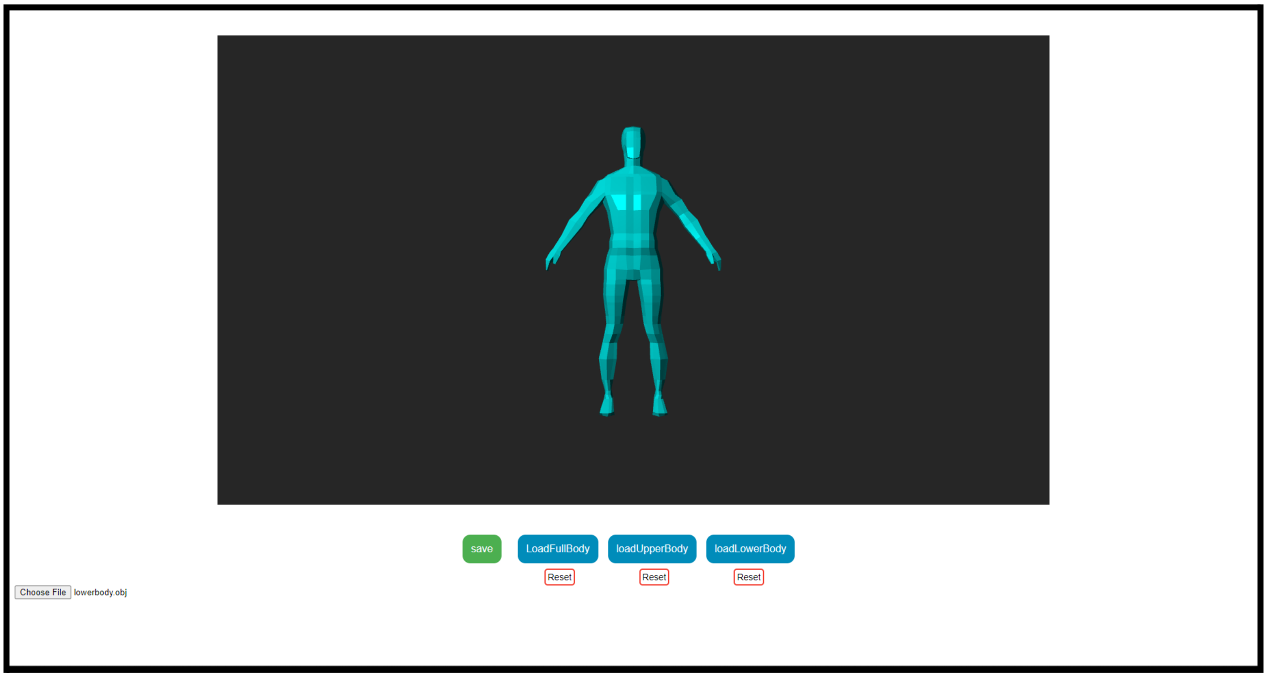
A web based program that is able to successfully load 3d models and allow the user the ability to draw on the model to depict which area he/she has the pain/discomfort in. Our program allows the user to switch between 3 different models of the human body namely a simple full body model , a high resolution upper half of the body and a similar lower half of the body . We have also added the functionality of downloading the drawn image after completion .

Problems faced :

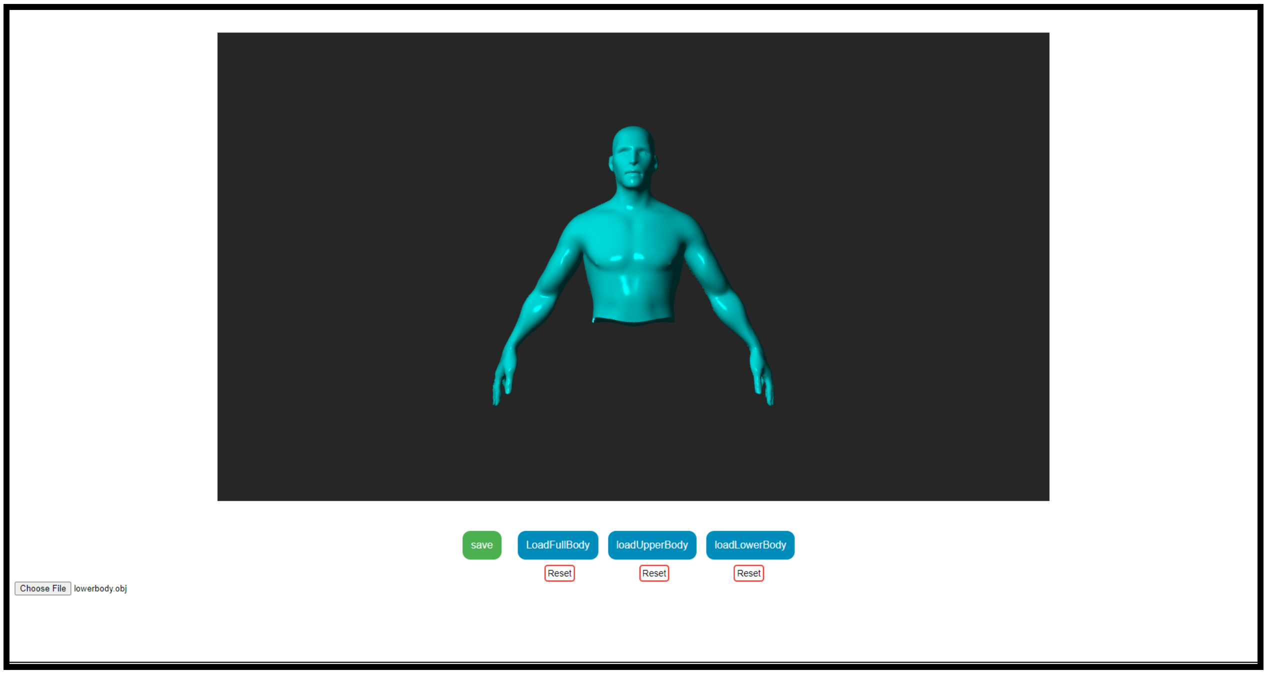
A problem that we faced while developing the model was

Finding a suitable 3d model : For the purpose of drawing on the model first we need to unwrap it into UV texture . This Uv coordinates are used for calculating the intersection of the drawing cursor on the 3d model by calculating the point of intersection of the ray and the object. For that the free available models were need to be unwrapped component by component. After unwrapping another problem that we faced was that due to the large size of the model we couldnt unwrap the whole object at once or else it created difficulties in draing on the obejct due to extremely close UV coordinates . Hence the Model was Divided in halves and each half was individually unwrapped.

Screenshots

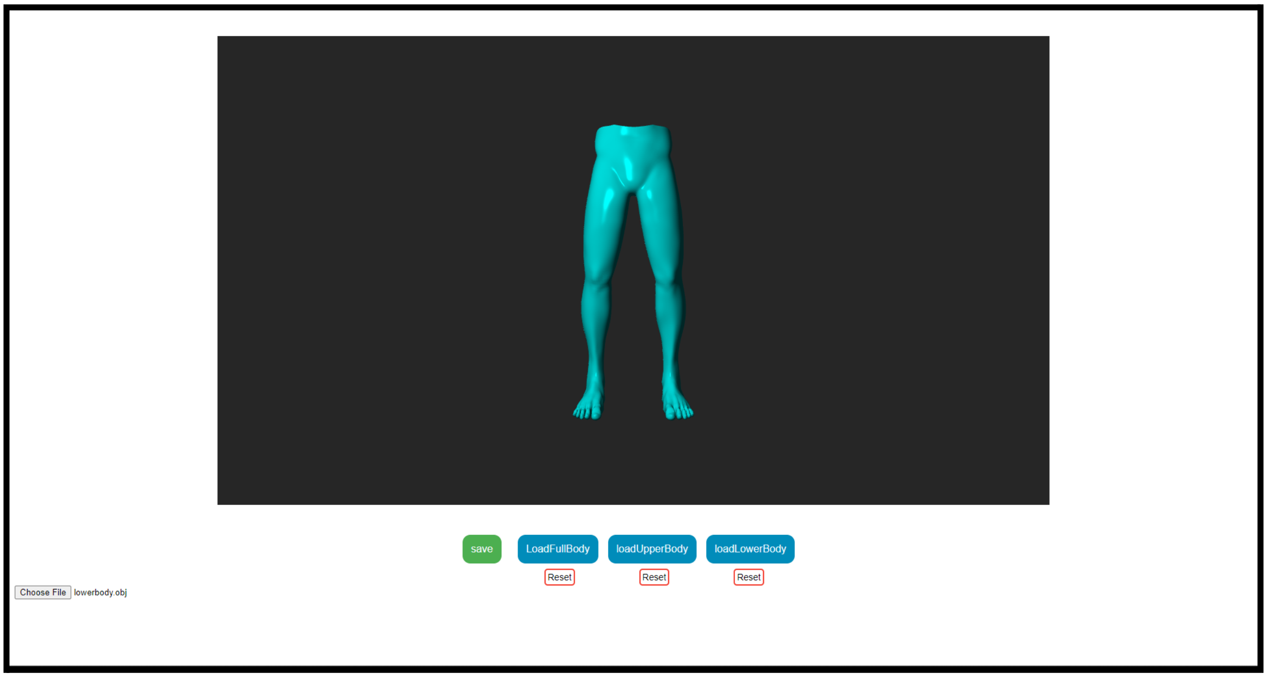


Home screen

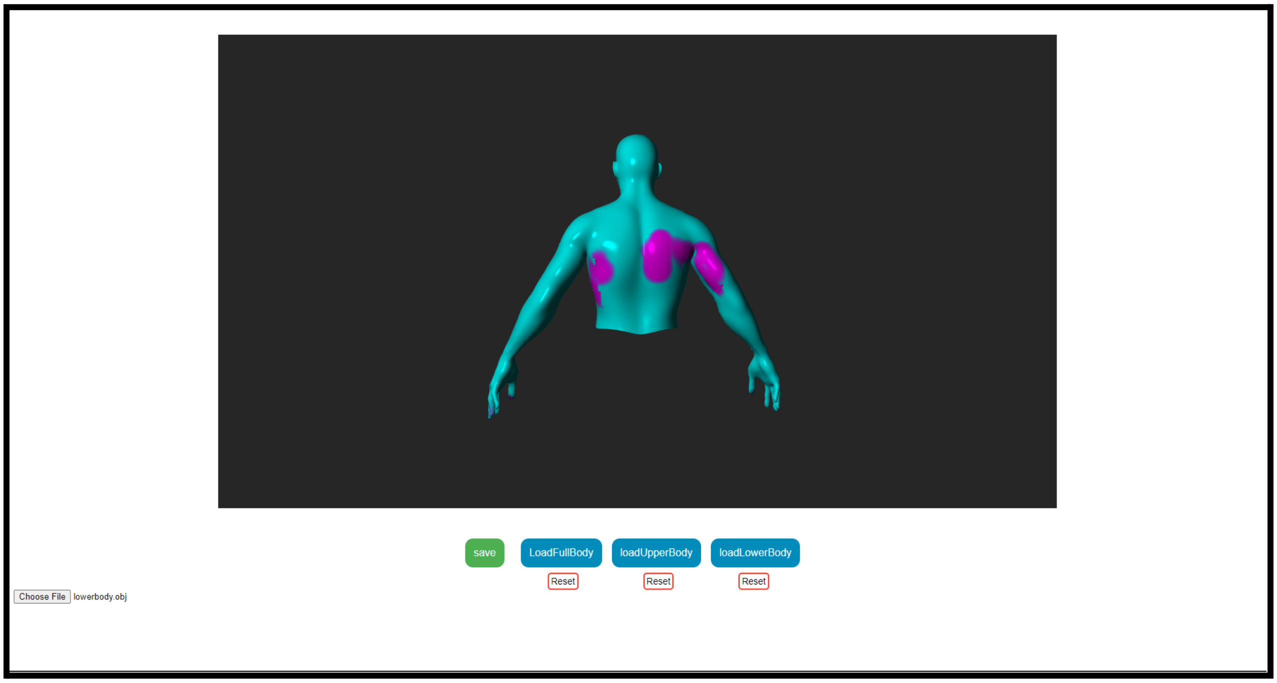


lowerbody.obj

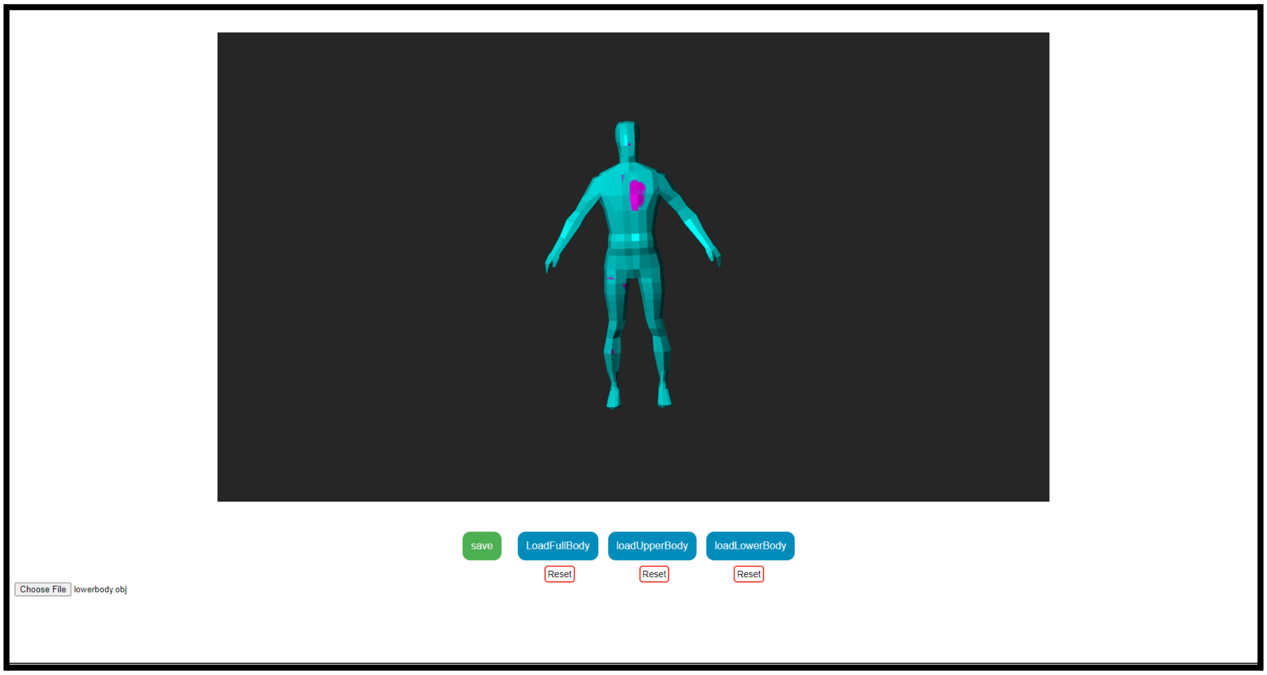
Upper body high resolution model



Lower body high resolution model



Example of drawing on the object



Another example of drawing on the object and rotation

Working and components

Basic working principle is that in order to draw on a 3d object we must first unwrap it and calculate its UV coordinates . Then When a cursor clicks on the model a ray is drawn from the origin to the model and the intersection point is calculated . This is then mapped to the UV space and the drawing takes effect. After that the model is rerendered with the updated UV space

1. Webgl.js : This file is for the initialization of the canvas and webgl variables
   a. clickTopaint : Checks if the cursor is in the canvas object or not and based on that given the location of the cursor coordinates it colors the object .
   It does that by creating a ray from the origin of canvas to the model and finding its intersection with it in UV coordinates .
   b. Start : This function is called at the beginning of the program when the canvas element is loaded . It checks for support of webgl on the browser and initializes the shaders and the geometry .After that it initializes the brush object and the main object . The default for it is the sphere .After this all the event handlers are added for the mouse .A periodic timer is initialized to call the DrawScene function.
   c. Drawscene : It first clears the canvas for drawing and initializes the texture space. After that We draw the object on the canvas.
   d. initshaders : this function is used for initialization of shaders for lighting of the model .

2. Brush.js : This module is for creating a brush object , with all the parameters such as intensity , size of the pointer and other library functions for binding the brush texture to the active texture unit of the 3d Model.

3. objObject : This contains the class for the 3d object and consists of functions such as loadmodel : which is used for parsing of the obj file and creating arrays for the vertices , textures and triangle faces , initbuffers : for initializing webGL buffer arrays, and intersection function which does the job of calculation of intersection between the ray and the object .