# Website Notes

## Tech-Stack

Database

Cloud: Azure Hosting Platform. Products utilized are Azure Cosmos DB (mongo), App Service (Free tier currently), and a Linux App Service Plan.  
To login to this portal use the following credentials:  
un - [bpowell@bioinvision.com](mailto:bpowell@bioinvision.com)  
pw - Cryo!Imaging781

## Completed Features

### Application Level

1. A login system has been created with the ability to denote users as clientele
2. Admin users can navigate to locked pages and see all of the active datasets

### Pre-processing

1. A python script has been created for this operation in which a user needs to create an input text file with a few parameters to generate the content needed for the web application.
2. The volume rendering has limitations due to the size of the texture data, and the basis file format not yet being available for 3D data. Therefore, a scaled down zip directory of png images is generated for this visualization. To mitigate the lower quality, when clipping the volume, the faces are pulled from the basis data and are 2D planes in the 3D scene.

### Interactions

1. Three dynamic scenes from the viewpoints of xy, xz, yz. All scenes cameras are synced in that the zoom level will be the same, and background color which can be altered by the user.
2. The most efficient data type has been selected for the 2D slices that allows a negligible lag experience when viewing and shifting slices without using tiling. Highest resolution tested has been 8192px x 4096px, which should be large enough to avoid a multiresolution approach.
3. Interacting with the image data, one can get the pixel value, measure a distance between two points, and change the alpha threshold of the image to remove unwanted artifacts such as leftover embedding material or hair.
4. A user can make an annotation on the XY plane. Users are then allowed to show/hide annotations, move annotations around, and change the font scaling of said annotations.
5. A user can make a view which saves all the scene data and is able to replicate what was on their screen when they pressed save.
6. A user can currently draw a mask (line loop) in the XY scene and save it to the database.
7. The XY scene can be downloaded as an image to your local machine by utilizing the export tab. The images have been restricted to a few selectable sizes while respecting aspect ratio.
8. An info tab has been provided to reflect the metadata important to the data set, such as PI and voxel size, and section thickness.
9. Clicking in any of the 2D scenes will redraw crosshairs to show where you are at in the dataset and update the corresponding scenes. This will also update the clipping inputs for the volume rendering, but the volume rendering currently does not have the same inverse relationship.
10. A content tab is provided to upload additional content, such as image stills and movies.
11. A bounding box option is present on the volume rendering to help with understanding the 3D context.
12. An opacity slider exists, and this will prove to be helpful once the cell detection texture data exists.

## Requested Features ( Title | Percentage Complete )

1. **Mask Renovation – 25% (2hours most changes | 4 hours for the interpolation) = 1 day**  
   Completed: Drawing and saving a mask with a name and other metadata.  
   Not completed: Change the name input to stay until a series of masks are completed. Add a button to preserve the mask for the next slice. Add a button that will interpolate between a starting and ending slice of masks. Also, provide a feature to show saved masks.  
   Notes: Explore using this mask data as a lookup for the 3D rendering, very similar to #3. This data can then grayscale/colormapped.
2. **3D side by side page ( CryoViz by MR or other modalities ) -- 75% = 2 day**  
   Completed: All the code to load each datatype in the most efficient way has been completed. Code for displaying the second Volume rendering will just be copied and camera’s will be synced.  
   Not completed: Just putting all the pieces together.  
   Notes: Keep as an option, pop it up and show same camera positions and comparisons that way
3. **Volume Rendering has cell detection option -- 60% = 3 Days**  
   Completed: Research on how to implement this and the necessary data structures and data types. Also discovered two separate object geometries will not work, and all the texture data needs to be synced to the same shader.  
   Not completed: Shader work needs to be finished. I attempted two different objects, but this confused webgl and it would choose an object to be drawn first for some very undesired results.  
   Notes: