

# User Manual

May 9, 2024

Version 1.0



**Sponsored by:** Dr. Elise Donovan, Sneha Vissa, Adonna Rometo

**Mentor:** Tayyaba Shaheen

**Team members:** Dayra Quinonez, Nicole Sylvester,  
Rino De Guzman, Bailey Rosato

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Installation</b>	<b>3</b>
<b>3. Configuration and Daily Operation</b>	<b>6</b>
<b>4. Maintenance</b>	<b>8</b>
<b>5. Troubleshooting</b>	<b>9</b>
<b>6. Conclusion</b>	<b>9</b>

# 1. Introduction

We are pleased that you have chosen the Diversified Anatomy and Physiology Lab Resource Application for your business needs. The Diversified Anatomy and Physiology Lab Resource Application is a powerful study tool for tailoring virtual three-dimensional anatomical models that have been custom-designed to meet your needs. Some of the key highlights include: adjusting skin tone, biological sex, and body size; the ability to zoom, rotate, and export; and a whiteboard to annotate. The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the Diversified Anatomy and Physiology Lab Resource Application in your actual business context going forward. We aim to make sure that you can profit from our product for many years to come!

## 2. Installation

As part of the final delivery, this section will go over the steps necessary to set up the product for future deployment and provide context for future developers. For the development of the product in the future, these are the steps for re-installing the product to be changed.

The basic steps for viewing the current application locally are as follows:

*These steps were done on a Windows 10 system using the Command Prompt System.*

### 2.1 Download the code

For the most recent version of the code, clone this repository.

```
$ git clone https://github.com/baileyrosato/Anatomia.git
```

Once the codebase is cloned on the device, run the following command:

```
$ cd anatomia
```

### 2.2 Download the necessary packages

To run the current iteration of this project, users will need to download the following packages:

- a. To download Node.js and npm, use the download link on the Node.js website: <https://nodejs.org/en/download>. Choose the correct download based on your correct operating system. The version used for this project is 20.9.0, with the 10.2.3 version of npm.

To ensure that the correct version of the software is installed, run these commands on your device:

```
$ node -v
```

```
$ npm -v
```

- b. Once npm and Node.js have been installed, you should now be able to install the Firebase database tools using npm with the following command:

```
$ npm install -g firebase-tools@latest
```

- c. Next, to ensure that the application can run, all the following dependencies must be installed. Begin each command with **\$ npm install** and then enter the following package names with their corresponding versions:

- **@emotion/react@11.11.3**
- **@emotion/styled@11.11.0**
- **@mantine/core@7.5.1**
- **@mediapipe/tasks-vision@0.10.10**
- **@mui/icons-material@5.15.7**
- **@mui/material@5.15.10**
- **@react-three/drei@9.97.0**
- **@react-three/fiber@8.15.16**
- **@tldraw/tldraw@2.0.0-beta.4**
- **dat.gui@0.7.9**
- **file-saver@2.0.5**
- **html-react-parser@5.1.10**
- **html-to-image@1.11.11**
- **html2canvas@1.4.1**
- **rc-slider@10.5.0**
- **react-dom@18.2.0**
- **react-router-dom@6.22.0**
- **react-scripts@5.0.1**
- **react-three-fiber@6.0.13**
- **react@18.2.0**
- **three@0.158.0**
- **web-vitals@2.1.4**

### **2.3 Connect the database to the application**

To connect the database to the application, you will need to place the config.js file in the src/Database folder of the application. The config.js folder is provided in the Google Drive provided to our clients.

**2.4 Run the application:** To view the application locally, users will use the ‘npm start’ command within the git repository.

```
$ npm start
```

**2.5 Hosting the application:** To have the application accessible from the web, our team has chosen to host the web app with Firebase Hosting services. To begin the hosting process, you must first be logged in to the Firebase account that contains the project. Our team has provided a Gmail account that the clients and future developers will have access to.

**\$ firebase login**

Now that you are logged in, create a build layer of the application. This command is compressing the application, so it may take a few minutes to finish running.

**\$ npm run build**

Next, you will now need to choose the Firebase project that will be used for hosting. Run the following command:

**\$ firebase init —project diversified-lab-resource-app**

The Firebase CLI will prompt with a confirmation of using the project, to which you will respond yes.

Next, there will be a list of options for which Firebase features to use. Choose the “Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys”. It will then ask a series of follow-up questions. The answers are provided below.

```
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
? File build/index.html already exists. Overwrite? No
i Skipping write of build/index.html
```

Next, run the **\$ firebase deploy** command.

Once Firebase is done with organizing files as it needs, it will provide a hosting link, which can now be visited from a web browser.

## 2.6 Testing

For the sake of future development, this section will outline how to run tests using the React Testing Library with the Jest framework. The GitHub provided to our clients will contain testing branches that have the necessary file structures that can be used as a reference. Before running the tests the following dependencies must be installed:

```
$ npm install @testing-library/jest-dom@5.17.0
```

```
$ npm install @testing-library/react@13.4.0
```

```
$ npm install @testing-library/user-event@13.5.0
```

Once tests are ready to run, they can be run from the command line in the project directory.

**\$ jest**

Running this command will give you a comprehensive list of test suites with a pass or fail status, and will include a breakdown of specific tests that have either passed or failed.

## 3. Configuration and Daily Operation

Currently, the application is deployed. However, certain tasks are essential for maintaining and operating the product efficiently. This section outlines the configuration steps and daily operations required to keep the product running smoothly.

### Firestore Configuration

#### 3.1 Firestore Console

The Firestore console is a web-based interface provided by Google Firestore that allows users to manage their Firestore projects. It serves as a central hub for configuring and monitoring Firestore provided services. You can access the Firestore console at this link:

<https://console.firebase.google.com/>. Then you are able to login with the provided credentials in the shared Google Folder. Once logged in, there will be a list of your Firestore projects. Select the Diversified Lab Resource App. Once selected, you will be directed to the project overview page where you can access Firestore Services such as Realtime Database, Storage, and Hosting. Each will be discussed further in the next sections.

#### 3.2 Firestore Realtime Database

Using the project shortcuts panel on the right side of the window, select Realtime Database. The Firestore Realtime Database is a NoSQL database designed to store and sync data in real-time between the web application and the cloud. Using the tabs at the top, you can navigate through the Realtime Database.

##### Data Tab

In the Data tab, you will find the data stored in the Realtime Database. Currently, 'courseData' is the top level collection that contains all of the documents needed for the web application. By clicking the down arrow to expand the 'courseData', you can view the units. By expanding a unit, you can view the model files and subunits. Each subunit contains fields for a title, description, and optional images.

##### Rules Tab

Moving to the Rules tab, these are the security rules for read and write access to the database. These rules ensure that only authorized users can access or modify the data. Currently, the rules are set to true for read and false for write. These can be updated by simply clicking in the rules box and adding your changes. Once done, be sure to save your changes.

##### Backups Tab

Currently, the Backups tab is not being used as it involves upgrading to a paid plan.

##### Usage Tab

Lastly, the Usage tab is used to monitor the database usage and track performance. It provides Billable Metrics that show the current month's activity and tracks free usage. As long as the usage stays within Free range, the project remains no-cost. You can also view load and rule metrics to monitor any performance issues.

### **3.3 Firebase Storage**

Using the project shortcuts panel again, select Storage. Firebase Storage provides scalable, secure, and efficient storage for content such as images, videos, and files.

#### **Files Tab**

In the Files tab, you will find all of the images and model files needed in the web application. From here, you can also upload files by pressing the upload file button. You can also open or delete files by checking the box next to the desired file and then choosing to delete or open.

#### **Rules Tab**

The Rules tab configures security rules to manage access permissions for storing and retrieving files. Similar to the Database Rules tab, you can update the rules by typing in the rules box and then saving your changes.

#### **Usage Tab**

Lastly, the usage tab shows Billable Metrics including storage and objects. This tab is used to monitor storage usage to ensure optimal performance and avoid exceeding storage limits. The storage limit is 5GB on the no-cost plan.

### **3.4 Firebase Hosting**

Firebase hosting provides fast and secure hosting for Firebase projects. By selecting the Hosting tab under project shortcuts, you can view the hosting dashboard.

#### **Dashboard Tab**

From the Dashboard, you can view the current release of the web application being hosted as well as any previous releases. You can also view the current domains that the web application is accessible from. You should use this tab to regularly review hosting settings to ensure the application is deployed correctly and accessible to users.

#### **Usage Tab**

The Usage tab allows you to monitor hosting usage and performance to address any issues related to scalability or resource consumption.

### **3.5 Rehosting**

If you wish to make changes to the application, you must rehost the website to see those changes live. Before making any changes to the product, ensure that you have the required development environment set up on your local machine. This includes installing any dependencies to match the production environment. To make changes to the web application, make them locally on your code branch. Retrieve the latest version of the source code from the project repository, make the desired changes to the codebase locally, and test your changes to ensure they function as expected.

Once you have completed and tested your changes locally, you can rehost the web application to make your updates live. To rehost, you will need to rebuild and redeploy the web application. Follow the steps for building and deploying outlined in section 2.5.

Test the deployed product to confirm that your changes have been successfully implemented by accessing the domain where the web application is hosted. By following these steps, you can

make changes to the application locally and rehost it to ensure your updates are applied smoothly.

## **Content Management**

In this section, we will cover tasks related to managing content within the database to ensure the product can be expanded to cover more content in the future. First, use the provided data initialization script 'dataInit.js' in the codebase to manage content stored in the database. This script allows you to upload new data, update existing entries, or delete data. Once you have made your desired changes to the 'dataInit.js' file, simply run the command 'node dataInit.js' to apply your changes to the Firebase Realtime database. Be cautious when making changes to the database to avoid errors.

If you would like to update files or images in the web application, you can make your changes within the Firebase Storage Dashboard by following the Firebase Storage File Tab instructions provided in section 3.3. If you are adding, for example, a new subunit image, you must upload the image to Firebase Storage. Then click on the file name and copy the storage location link that starts with 'gs://'. Then in the images field in the 'dataInit.js' file, you can paste that link so that the database correctly retrieves and displays the file in Firebase Storage.

## **4. Maintenance**

Fortunately, due to the nature of the application, the effort required to maintain the application is relatively low. To ensure the continued functionality of the Diversified Anatomy and Physiology Lab Resource application, certain maintenance tasks can be performed regularly. Below are the key activities to undertake to maintain the health of the product.

### **Dependency Updates**

Regularly review and update dependencies to incorporate bug fixes, security patches, and new features. Check for updates to the libraries and technologies such as React.js, Three.js, MaterialUI, and Tldraw.

### **Source Code Access**

Access to the application's source code is essential for future development and troubleshooting. A local copy of the repository on a USB stick will be given to the clients for backup purposes. In addition, to ensure access to the GitHub repository, the clients are given a Google Drive with a GitHub account containing the source code of the project, a Firebase account with configuration for the database, and a Google Email. All the account and login information is accessible within the Google Drive folder shared directly with the clients. To prevent unauthorized access, the information will not be shared in this document.

## 5. Troubleshooting

There may be instances where developers encounter challenges during installation, configuration, or daily operation. Below, we've compiled a troubleshooting guide to address possible issues that the developer may experience with the application.

### Installation Issues

#### Incompatibility with Node.js and npm versions

Ensure that you have installed the correct versions of Node.js and npm as specified in **section 2.2**. The installed versions can be verified by running '\$ node -v' and '\$ npm -v' commands. If versions are incorrect, reinstall Node.js and npm to match with the versions specified in **section 2.2**.

#### Connection and Hosting Issues

Ensure that the 'config.js' file is correctly placed in the designated folder ('src/Database'). Confirm that the Firebase project is set up correctly, by verifying the Firebase credentials and project settings. Check for any errors in the deployment process and refer to Firebase documentation for additional troubleshooting steps.

#### Course Content Management Issues

If you are having trouble managing content within the database or updating files/images in Firebase Storage, verify that the 'dataInit.js' script is correctly configured and formatted. Sometimes single (') or double (") quotations in a content field may disrupt the formatting of the stored data in the database. Ensure that Firebase storage rules are followed when uploading, updating, or deleting files. It is also important to ensure proper permissions and storage location references to allow the correct redirection of data

## 6. Conclusion

With best wishes from Team Anatomia: Dayra Quinonez, Nicole Sylvester, Rino De Guzman, and Bailey Rosato. It has been a pleasure working together in creating this impactful application. While we are all moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization. **Our emails: Dayra Quinonez ([dayraquinonez11@gmail.com](mailto:dayraquinonez11@gmail.com)), Nicole Sylvester ([niccisylvester@gmail.com](mailto:niccisylvester@gmail.com)), Rino De Guzman ([rinodeguz@gmail.com](mailto:rinodeguz@gmail.com)), and Bailey Rosato ([baileydawn2010@gmail.com](mailto:baileydawn2010@gmail.com)).** Team Anatomia cannot wait to see how the application progresses and how far it goes.