1. Making accounts
    a. If you don't already have a Github.com account, make one. Using Github is pretty important to the project.
    b. Also, you'll need an account to use firebase with. I believe a gmail account will do, I'm not exactly sure if you can use another type with it. But at least you'll need to be able to create a firebase project at https://firebase.google.com/
2. Setting up the repository
    a. Create your own repository on Github. Upper right hand corner, + sign, New repository.
    b. The best way to set it up would be to duplicate the repository without forking it. To do this, follow the instructions found here: https://help.github.com/en/articles/duplicating-a-repository
    If you find using the Terminal or Git Bash confusing, download the repository from here: https://github.com/DrSharky/VTMSite by clicking on the "Clone or Download" button, then clicking Download ZIP. Open the .zip and remove the .git folder if there is one. Then you should be able to push the files that were inside the VTMSite-master folder you have to your repository, and it should be a duplicate without forking.
    c. Once you have your files uploaded, go to the "settings" tab for your repository, and create a GitHub Pages site for it. This will be the host for the website. Don't choose a theme, and make sure the Source is set to the master branch. It will create a new branch called gh-pages branch.
3. Setting up Firebase
    a. It may seem like you're done, but no, not yet. So far we've duplicated the site, but it's still connected to the original VTMSite database. Back to firebase. Go to https://console.firebase.google.com/. You can also click "Get Started" on the firebase site if you still have it open.
    b. Click "Add Project" and name it whatever you like. On the next page you'll see something about analytics. I don't use it, and the site doesn't require it. Skip it for now, you can set it up later if you want.
    c. It'll take a few seconds to create your project. Once it's done, you'll see a project overview page. There are a few icons you can click on. Right above where it says "Add an app to get started", click on the </> icon to start setup with your web app.
    d. Again, name your web app whatever you want, and click "Register app".
    e. Once you've done that, copy the code that it shows where it says "Your web app's Firebase configuration"
    f. Locate the "index.html" file inside the project files you downloaded, in the VTMSite project folder & open it with a text editor.
    g. Find & replace the firebase configuration code that is in there with the code that you just copied from Firebase. Push & upload your changes to the index.html file to Github, and now your site should be connected to your Firebase project.

4. Configuring Firebase Authentication
    a. Yeah, still not done… Now you need to configure Firebase to contain actual data. First, on the left side of the console overview page, click on “Authentication”
    b. Click on the “Setup sign-in method” button, click on “Email/Password”, enable that option, and then save it.
5. Creating a database
    a. On the sidebar of the Firebase admin, click on “Database”. Then click on the “Create Database” button in the center of the page.
    b. A window should show up, and will give you 2 options regarding the rules about reading/writing to your database. For now, select “Start in test mode”.
    c. It will then ask you to select a server location for your database. This is arbitrary, just pick what you think will best work for you.
6. Configuring the database
    a. I set up my database before Firebase created their “Cloud Firestore” system, so I’ll continue the guide on setting up the Realtime database system that I have. I might update this later when I get around to figuring out the new firestore feature.
    b. It will default to creating cloud firestore, so at the top next to where it says Database, switch Cloud firestore to “Realtime Database”.
    c. Now we set up the rules, so not everyone can read & write to your database. I have my rules set so that only registered users can read from & write into their own entries in the database. To get started on it, click the “Rules” tab at the top of your database page.
    d. Delete whatever is in the boxed in area, and enter the following code:

```
{
  "rules": {
    "characters":{
          "$uid": {
          ".read": "$uid === auth.uid",
          ".write": "$uid === auth.uid"
          }
    },
    "characterNames":{
      "$uid":{
        ".read": "$uid === auth.uid",
          ".write": "$uid === auth.uid"
      }
    }
  }
}
```

    e. Press the “Publish” button that appeared.
    f. That’s it, the database is configured. There’s no data inside the storage, but you can check it later once you’ve saved a character, and you’ll see your saved data.

7. Testing the Site
    a. I thought I'd add this just in case. I like to use a quick way to test the changes I've made to the site, so maybe you'll find it useful.
    b. For my testing, I use a handy tool called http-server to locally host the website, so I can test changes before uploading them. This is optional, but if you'd like to test this way, download & install Node.js: https://nodejs.org
    c. During the install process, just make sure that the "npm package manager" and "Add to PATH" are both selected to be installed. You'll see them come up during the install.
    d. Once that's done installing, open up a console window (type cmd in the search bar, or bash on macOS). Now we can use the npm command to install http-server.
    e. In the window, enter the following command: npm install http-server -g
    f. After http-server is installed, you can close the console window.
    g. Hold shift and right-click on the folder where the website project is located. You should be able to click on an option called "Open PowerShell window here".
    h. With the powershell window open in the project folder, type: http-server -o
    i. The site should now be running locally on your computer.
8. Done
    a. That's about all there is to it. You should now be able to test the site, register your username for a new account, save a character, and then check your database to see the new data entry.

This is a first draft of a guide for setting up the VTMSite project. I'm sure there are flaws, errors, notes I should make, etc. Feedback is always appreciated.