**This doc, if you want to make same project from scratch instead of cloning and running the project directly.**

1. Installed Ganache by downloading dmg from it’s official website and installed manually.(It is used for the private blockchain network on your PC)

2. Installed truffle using cli command “npm install -g [truffle@5.0.2](mailto:truffle@5.0.2)” on the PC. (Truffle is like framework like Laravel for blockchain projects)

3. Create directory for the project.

4. Go to the current directory and using CLI command “truffle init” get all the required packages inside the project.(we can use CLI command truffle version before to check the version of truffle as well as to check if truffle is installed properly)

5. Created package.json get the reference using the reference projects and after saving the file executed CLI command “npm install” to install all the dependencies packages for our project we saved in the package.json

6. Created pet\_adoption.sol contract in contract directory

7. Go to truffle-config.js and set the host and port of the running on the ganache application. (Code referred from the online source) basically it will be a connection between our application and local blockchain network running on ganache.

8. Go to migration directory and create a 2nd migration file to deploy our smart contract that we created in the previous step. After that go the terminal and execute “truffle migrate”(this is very similar to that we in asp.net mvc or Laravel where we have a scheme file and we don’t need to go again repeatedly to DB to update tables or fields, so we just update those schema and finally execute migrate which will automatically change/update the state of our database. However over here we have our smart contract which we will need to deploy or migrate to the blockchain)

9. Creates src folder which includes all the CSS, JS(app.js), images, index.html and also the json file which includes the data of pets and which will be used to fetch pets data.

10. Set bs-config.json(browser sync) file which is used to set the routes.

11. Connect metamask and ganache: select custom RPC-> In Network name write private network -> New RPC URL write <HTTP://127.0.0.1:7545> -> chain id write 1337 -> click save.

After that make sure private network is selected on the top right hand side-> click on logo next to private network then click import account -> select private key and then go to Ganache window on top account there is key symbol which will give private key copy that and paste it in import account dialog box. We can see now our top account in ganache is available in metamask and we can see the etherium coins which are available for that account.

12. Set the index.html and app.js as the reference code and if you have already migrated then execute “npm run dev”.

13. It will execute the page and once we click on adopt button it should redirect to metamask popup ask for the permission. (Make sure the imported account is connected to the running site if not then try to click on three-dot burger icon next to account and click connect sites and it will ask to connect account to the current localhost site. Once connected the whole project should run fine.)

Pets photo reference:

1. corgi1: <https://favim.com/image/7073677/>

2. retriver2: https://hypeauditor.com/preview/tuckerbudzyn/

3. rest of the animals’ photo was taken from https://www.goodhousekeeping.com/life/pets/advice/g1236/animal-instagrams/?slide=19