Vue.js Frontend Deployment on Azure

Step 1: Create a frontend deployment slot on Azure portal:

- Go to the App Service you already created for the backend
- Select 'Deployment Slots' option from the sidebar on the left, then click 'Add Slot' to create a new slot for your frontend
- Fill out the dialog like this
  - `Name:` <your_backend_app_name>-frontend
  - `Clone Settings From:` Select 'Do not clone settings ' from the dropdown menu
- After azure has finished creating the slot, go to the apim for your frontend by clicking 'API Management' in the sidebar
  - There shouldn't be any apim linked at this point - if there is then something went wrong
- In the API Management menu, fill out the fields like this
  - `API Management` Select the same apim that also linked to your backend slot, should be named something like <your_backend_service>-apim (MAKE SURE THEY ARE THE EXACT SAME)
    - if your backend apim isnt there, something went wrong
  - `API` Select the same API you've been using for the backend
- Now that you're in the apim window with the apim linked, select the 'APIs' option from the sidebar
  - You should see all the same API definitions that were there before
- Scroll all the way to the right to the 'Backend' tile and edit the HTTP(s) endpoint
  - Enter the exact URL of your backend service and save everything

---

Step 2: Build and Deploy your frontend

- Open the /Frontend folder in VS Code **(NOT VISUAL STUDIO)**
- Open a new terminal and navigate to the /Frontend folder
- Run `npm run serve`

    - If you get the error `code: 'ERR_OSSL_EVP_UNSUPPORTED'` run the command `export NODE_OPTIONS=--openssl-legacy-provider` to fix it, then run `npm run build` again
- Now find the frontend production folder that was generated (should be located in /Frontend/dist)
- Before continuing, you must have both the' Azure App Service' and 'Azure Account' extensions installed on your VSCode instance
    - Follow the installation directions and sign into your Azure acct
- Manually configure Azure to publish to the deployment slot you created earlier by running the command: `az webapp config appsettings set --resource-group <you_resourcegroup> --name <your_deployment_slot_name> frontend --settings PROJECT="<frontend>/<frontend>.esproj"` while replacing the first two < > fields with your data (DO NOT change the <frontend> field at the end)
- Now find the Azure extension button on the bottom of the left toolbar with the solution explorer (should be on the very bottom) and select it
- In the dropdown menu that appears, navigate to the App Service that holds both your deployment slots
- Now right click on the frontend slot you made and select 'Deploy to slot...'
- This should prompt a popup at the top of your VScode window asking you if you want to deploy using the the current folder (/Frontend) - **DO NOT SELECT IT**
- Instead, select the 'Browse' option underneath it then select the /dist folder your generated earlier and deploy from there
- After successfully deploying , your frontend should now be live at the URL set by your frontend deployment slot

---

Step 3: Final Steps/Troubleshooting

- Refresh everything on Azure
- Make sure the API definitions still work when tested in the new deployment slot
- If your live frontend still isn't successfully linked to your API, ensure you've done all of the following in the Frontend slot
    - Set the Connection String for your database (must be the exact same as the connection string used in the other slot

    - Make sure the checkbox for the 'Deployment Slot' option is UNCHECKED
        - Must also be done for the connection string in the backend slot
- Enable CORS (must also be enabled in backend slot, if not already