1. You may be curious about what information is stored in this file. So please use **cat** to display the content in "RealEstate.csv " using a relative pathname.

cat RealEstate.csv

1. We know that cat is good for showing the content of a small file. But since the file contains many lines, maybe you still cannot find out what information this files stores after step (1). So please use **head** to list the first three lines in "RealEstate.csv".

head RealEstate.csv

1. Use wc to check the number of homes sold out in "RealEstate.csv ".

wc RealEstate.csv

1. Finish the task in step (3) by using the cat command.

cat -n RealEstate.csv

1. Use mkdir to create a new directory "public" under your own home directory

using relative pathname.

mkdir ../public

1. Copy "RealEstate.csv" into your "public" directory and name it as "myRealEstate.csv".

cp RealEstate.csv /home/anguyen117/public/myRealEstate.csv

or with relative path:

cp RealEstate.csv ../public/myRealEstate.csv

1. Display the absolute pathname for current working directory.

Using the command “pwd”, we get the absolute pathname for the current working directory displayed below:

Input:

pwd

Output:

/home/anguyen117/Lab2\_P2

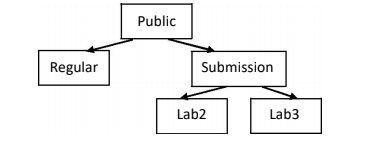
1. Check the existence of "myRealEstate.csv" using ls with an absolute pathname.

ls /home/anguyen117/public

1. Go into your "public" directory using relative pathname.

cd ../public

1. Use mkdir to create a file structure as below in your "Public" directory using relative pathnames.



mkdir Regular Submission Submission/Lab2 Submission/Lab3

1. Rename the directory "Regular" as "Others".

Using relative pathnames:

mv Regular Others

Using absolute pathnames:

mv /home/anguyen117/public/Regular /home/anguyen117/public/Others

1. Use cp to copy directory "Lab2\_P2" from your home directory to "Lab2"

using relative pathname.

cp -R ../Lab2\_P2 Submission/Lab2

1. Remove the directory "Lab2\_P2" which locates at your home directory.

rm -r ../Lab2\_P2

1. Use history to list the commands you previously typed.

history

1. Store the last 50 commands you typed neatly into a file "Lab2\_2.txt", one

command per line and submit it in Google Classroom.

106 mv Cookies Submission

107 ls

108 cd Submission

109 ls

110 cd ..

111 cd ~

112 ls

113 cd public

114 cp ../Lab2\_P2 Submission/Lab2

115 cd ~

116 ls

117 cd public

118 ls

119 cd Submission

120 ls

121 cd Lab2

122 ls

123 cd ../..

124 cp -R ../Lab2\_P2 Submission/Lab2

125 cd ~

126 ls

127 cd public

128 ls

129 /cd Submission

130 cd Submission

131 ls

132 cd Lab3

133 ls

134 cd ..

135 cd Lab2

136 ls

137 cd Lab2\_P2

138 ls

139 cd Lab2\_P2

140 cat Lab2\_P2

141 cd ..

142 ls

143 cd ..

144 ls

145 cd ..

146 cd ~

147 ls

148 cd Lab2\_P2

149 ls

150 cd ../public

151 rm -r ../Lab2\_P2

152 cd ~

153 ls

154 cd public

155 history

I input a lot of commands to check the results of the answer commands made for each step to check if it was the correct command.

(Also made a file in Unix with the last 50 commands called “Lab2\_2.txt”)