```
#!/usr/bin/perl

# ensure all fatals go to browser during debugging and set-up
# comment this BEGIN block out on production code for security
BEGIN {
    $|=1;
    print "Content-type: text/html\n\n";
    use CGI::Carp('fatalsToBrowser');
}
use strict;
use CGI;
use CGI qw/:standard/;
use lib "Perl-modules";
use XML::Simple;
use HTML::TreeBuilder;

my $simple = XML::Simple->new( );  # initialize objects
my $cgi = new CGI;
my $xmldirectory = "../htdocs/smartpost/xml_files/";
my $files = dirlist($xmldirectory);
my @jobs;
my $baseurl = "http://02e8a4f.netsolhost.com/smartpost/jobs/";

# print "Content-type:text/html\n\n";   # required header for web output

## create job postings from xml files
foreach my $fileInDirectory (@$files) {
    my $tree = $simple->XMLin( '../htdocs/smartpost/xml_files/' . $fileInDirectory );   # read, store
        document
    my $name = $tree->{ id } . '.html';
    ## load global hyperlink array with: Company + ID + Title
    push(@jobs, a({-href="$baseurl$name"},"$tree->{ company } - $tree->{ id } - $tree->{ title }") .
        '<br />');
    open (MYFILE, '>>' . $name);
    print MYFILE $cgi->start_html(  -title => $tree->{ id } . ' - ' . $tree->{ company } . ' - ' .
        $tree->{ title },
                                    -meta => {  'keywords' => $tree->{ job } . ', job',
                                                'description' => 'job description'},
                                    -style => { -src => '/css/main.css'} ) .
                                    '<strong>' . $tree->{ title } . '</strong>' .
                                    $tree->{ description } .
                                    a({-href="$tree->{ url }"},"$tree->{url}") .
                                    $cgi->end_html . "\n";

    # move files from cgi-bin to jobs folder and enable execute permissions # <-- uncomment for
        production
    rename($name, '../htdocs/smartpost/jobs/' . $name) || print "Lack permission to rename " .
        $name . " or file does not exist.<br><br>";
    chmod(0755, '../htdocs/smartpost/jobs/' . $name) || print $!;
#   rename('../htdocs/smartpost/xml_files/' . $fileInDirectory, '../htdocs/smartpost/xml_files/
    processed/' . $fileInDirectory); #move xml file
}

# create array of newly created hyperlinks and add to exisitng ones on index.html
my $oldlinks = '';

## sort the list of hyperlinks
my @sorted = sort { $a cmp $b } @jobs;   # ASCII-betical sort

# creates an array of xml filenames
sub dirlist {
        my $dir = shift;
        my @filelist;
        opendir(DIR, $dir) or die "can't opendir $dir: $!";
        while (defined(my $file = readdir(DIR))) {

                # We only want files
                next unless (-f "$dir/$file");

                # Use a regular expression to find files ending in .xml
                next unless ($file =~ m/\.xml$/);
                push(@filelist, $file);
        }
        closedir(DIR);
        return \@filelist;
}
```

This code loops over the array of file names

```
# # create array of job posting filenames
# sub joblist {
#         my $dir = shift;
#         my @filelist;
#         opendir(DIR, $dir) or die "can't opendir $dir: $!";
#         while (defined(my $file = readdir(DIR))) {
#
#                 # We only want files
#               next unless (-f "$dir/$file");
#
#               # Use a regular expression to find files starting with SP
#               next unless ($file =~ m/*.html/);
#                 push(@filelist, $file);
#         }
#         closedir(DIR);
#         my @sorted = sort { $a cmp $b } @filelist;   # ASCII-betical sort
#         return \@sorted;
# }

# # create list of formatted hyperlinks for job postings
# my $jobdirectory = "../htdocs/smartpost/jobs/";
# my $linklist = joblist($jobdirectory);
# my $joblinks = '';
# foreach my $jobInDirectory (@$linklist) {
#   next unless ($jobInDirectory =~ m/\.html$/);
#   $joblinks .= '<a href="http://02e8a4f.netsolhost.com/smartpost/jobs/' . $jobInDirectory . '">' .
    $jobInDirectory . '</a>' . '<br />';
# }

use HTML::LinkExtor;        # load the needed module

my $s = $baseurl . 'index.html';
print "These are the existing jobs on the webpage:\n";
my $p = HTML::LinkExtor->new(); # create a link extractor object
$p->parse_file($s) or die("\nOutput incomplete. Input file not found: $s\n");
foreach($p->links)  #  iterate over the hyperlink list and print each
{
    print "@$_\n";  # each list element is an array that must be dereferenced
}

## render job postings web page
open (LISTING, '+> ../htdocs/smartpost/jobs/index.html');
print LISTING $cgi->start_html( -title => 'Job Postings',
                                -meta => { 'keywords' => 'employment opportunities, jobs',
                                           'description' => 'job descriptions'},
                                -style => { -src => '../css/main.css'}, ) .
                                '<div id="main">' .
                                    '<div id="top" onclick="parent.location=\'http://
                                        www.hiretellers.com\'">' . '</div>' .
                                    '<div id="help"></div>' .
                                    '<div class="head">' . 'Active job postings:' . '</div>' .
                                    '<div id="links">' . "@sorted" . '</div>' .
                                    '<div id="valid">
                                        <p>
                                            <a href="http://validator.w3.org/check?uri=referer"><img
                                                src="http://www.w3.org/Icons/valid-xhtml10"
                                                alt="Valid XHTML 1.0 Strict" height="31" width="88" />
                                                    ></a>
                                        </p>
                                    </div>' .
                                '</div>' .
                                $cgi->end_html . "\n";

# message to user upon execution of this file
print "Done<br><br>";
print "New job(s) posted at: <a href=\"http://02e8a4f.netsolhost.com/smartpost/jobs/index.html
    \">http://02e8a4f.netsolhost.com/smartpost/jobs/index.html</a><br><br>";
print "Jobs added:<br />";
print "@sorted";
### end of script
```