# Niyuj Coding Challenge

Thank you for your interest in Niyuj. As part of our candidate evaluation process for positions involving significant amount of software development we ask candidates to complete a coding challenge. This challenge is a critical part of the candidate evaluation process. You are strongly encouraged to produce a response that is representative of your skill and experience.

## Problem Statement

Write a client/server sample
Platform: Linux
Programming Language: C

Write a sample client/server project. The server is a socket server listening on port 54321, it serves as a directory listing server which will list current directory content and send back to client. The client use socket to connect to the server, it supports 4 commands below. User will enter command in client and display result sent back from the server.

- **pwd**: display server current directory under the session
- **ls**: list files under current server directory. The display format will be:
    - o <type> <filename> <date>
        - type: file or dir
        - filename: name of the file/dir
        - File creation date
- **cd** <directory> : enter the directory
- **bye**: close the connection session

## Test Guidelines

- There should be no security loopholes in the implementation
- Guard against buffer overflow and use only safe string functions
    - ✔ Prefer snprintf() over sprintf(), prefer strlcpy() over strcpy() or strncpy()
- The server should allow for multiple clients to connect simultaneously
- Each client-server connection should be an independent "session"
    - ✔ Consider implication of the "cd" command
- Avoid obvious memory leaks in the server, at least in the happy path
- Ensure code is consistently indented
- Write a README file containing:
    - ✔ solution design
    - ✔ instructions to build and a sample run output
    - ✔ notable assumptions made in the solution
    - ✔ known limitations of submitted code
    - ✔ justification if any of the above guidelines are being skipped

**For bonus marks:**

- Avoid resource leaks in the server - even in error conditions
- Handle all errors such that the server does not crash
  - Log useful messages that could help with troubleshooting
- Handle directories containing a large number of files without proportionate memory consumption
- Fix all build time warnings (mention Linux distribution and gcc version used)
- Reduce code duplication between client and server

## Submission Instructions

· Make a directory under your name (eg. if your name is Ashish Netam, then make a directory ashish.netam).
· Put all your source and header files, Makefiles/Build scripts under this directory
· Create a tar or zip archive of this directory, and mail that as submission
· Make sure that you have tested your solution well before submission
· Submission via github or any similar channel repository won't be accepted
· Submission where all files (source, header, readme, build script etc.) are pasted separated by marker lines in a single text file, won't be accepted